ROBUST WATERMARKING USING COMPRESSED SENSING FRAMEWORK WITH APPLICATION TO MP3 AUDIO

Mohamed Waleed Fakhr

Electrical and Electronics Engineering Department, University of Bahrain, Manama, Bahrain
mfakhr@uob.edu.bh

ABSTRACT

In this paper a watermark embedding and recovery technique is proposed based on the compressed sensing framework. Both the watermark and the host signal are sparse, each in its own domain. In recovery, the L1-minimization is used to recover the watermark and the host signal almost perfectly in clean conditions. The proposed technique is tested on MP3 audio compression-decompression attack and additive noise attack. Bit error rates are compared with standard spread spectrum embedding. The proposed technique is implemented for both time domain and frequency domain embedding with a unified approach. The Walsh-Hadamard transform (WHT), the discrete cosine transform (DCT) and the Karhunen-Loeve transform (KLT) are compared in the host signal sparsifying process. Significant performance improvements in all tested conditions are achieved against the spread spectrum embedding. A payload as high as 172bps in additive noise attacks, 86bps in 128kbps MP3 attacks and 11bps in 64kbps MP3 attacks are achieved at small bit error rates and acceptable MP3 audio signal quality.

KEYWORDS

Compressed Sensing, Audio Watermarking, MP3 Audio, L1-Minimization, Sparse Signals.

1. INTRODUCTION

Digital robust watermarking embeds a private secure code within a multimedia file (e.g., MP3 music) where the watermark detection is possible only for the certified authority. The watermark should be imperceptible, secure, with high payload and robust to some attacks that do not destroy the original host signal. This is a truly challenging problem for audio signals, and in particular, for MP3 audio where MP3 compression/decompression attack is inevitable.

Traditionally, spread spectrum (SS) and quantization index modulation (QIM) based approaches have been used for watermark embedding and recovery, however, both suffer from problems when it comes to compression/decompression attacks, and additive noise attacks. In particular, SS suffers from host rejection problem which reduces its payload capacity to increase the spreading rate, while QIM suffers from low immunity to additive noise and both have problems with MP3 compression/decompression attacks. Recently, there has been a considerable amount of work in robust watermarking of MP3 audio to overcome some of the problems of the traditional

DOI : 10.5121/ijma.2012.4603

techniques [1-15]. Recent work focuses on the effects of MP3 compression/decompression sampling rate conversion, and synchronization attacks, among other attacks. New watermarking techniques have emerged which combine the histogram based embedding, SVD of transform domain coefficients and the QIM are reaching promising payload values with improved robustness to different attacks [1-15].

Compressed sensing is a relatively new signal processing framework in which a sparse representation of a signal can be recovered from its noisy compressed measurements using the L1-norm minimization [16-19]. More recently, it has been shown that a sparse signal and an additive sparse interference can both be recovered perfectly under some conditions [20-21]. This recent work has inspired the novel idea of a “sparse watermark” which is presented in this paper in the compressed sensing framework. The sparse watermark works perfectly under clean, no attack or "fragile" conditions, where the watermark and the host signal can both be recovered with no errors, giving a very good reversible fragile watermarking approach. However, since the aim of this paper is robust watermarking, the practicality of the proposed technique is tested for MP3 music with various attacks, and is compared to the traditional SS approach for time domain and frequency domain embedding.

The rest of the paper is organized as follows. Section 2 discusses the related recent work in MP3 audio watermarking. Section 3 gives a brief description of the related theory of compressed sensing and the recent work in the Justice Pursuit [20] and extended LASSO [21] techniques which inspired the proposed technique. Section 4 describes the details of the proposed sparse watermark embedding and recovery, and the method used for sparsifying the audio host signal. Experimental results and comparisons are shown and discussed in section 5 and finally section 6 gives the conclusions and future work.

2. AUDIO WATERMARKING RECENT WORK

In a recent paper [1] authors have made a comparison between most important MP3 audio watermarking research based on 7 criteria, namely, 1- the methodology being used (and that includes the complexity of the embedding and detection algorithms), 2- the imperceptibility of the watermark, 3- the robustness against attacks (e.g., compression/decompression, sampling rate conversion and de-synchronization), 4- the payload, 5- the security of the watermark, 6- whether the technique is blind or not and 7- if the algorithm and watermark are known to the attacker, can he use them to embed another audio media to frame the owner. Most of the recent techniques suffer from degraded performance against MP3 compression and sampling conversion. Payloads from as low as 2bps to up to 1378bps were reported with MP3 compression attacks (32kbps, 64kbps and 128kbps) with bit error rates in the ranges between 0.01% and 16% and SNR between 26 and 40dB. As shown in table 1 which summarizes some of the most recent approaches used for MP3 robust audio watermarking, ordered based on the MP3 attack quality.

Table 1. MP3 Robust Watermarking Summary

Algorithm	MP3 Quality	SNR	%BER	Payload
Dhavale 2011 [8]	32kbps	26dB	0.4	1378 bps
Dhavale 2012 [9]	32kbps	>40dB	16%	344bps
Noriega 2010 [2]	64kbps	40dB	0.013	230 bps
Bhat 2010 [3]	64kbps	>30dB	1	196 bps
Wu 2005 [12]	64kbps	30dB	0.043	172bps
Hamdouni 2012 [11]	64kbps	30dB	0.4	100 bps
Yang 2010 [10]	64kbps	30dB	0.02	22 bps
Ercelebi 2009 [13]	128kbps	30dB	0.5	170bps
Wang 2004 [14]	128kbps	30dB	0.06	11bps
Xiang 2007 [4]	128kbps	30dB	0.175	2bps

Most of the recent work reviewed here rely on the concept of quantizing some parameters following the quantization index modulation (QIM) approach [2,6,7,15]. The quantized parameters are in a transformed space, for example, the Walsh-Hadamard [8], or the wavelet domains [2,9,10,11,14,15]. Many of the recent techniques apply singular value decomposition (SVD) [3,6,7] as a means of making the embedding more robust. Spread spectrum embedding, patchwork algorithm and histogram modification of selected coefficients have also shown good results [4,5,6,10]. One criticism over most of the reviewed techniques is that embedding is done in a selected number of coefficients with quantization modulation approach. Those two factors make such techniques more vulnerable to attacks. In one hand, QIM is very sensitive to additive noise attacks. On the other hand, if the attacker knows which coefficients are used in embedding, the attack would be much more destructive to the watermark and less degrading to the host. One advantage of the technique proposed here is that embedding is highly distributed in the time domain, thus, highly robust to additive noise and selective attacks. It is concluded that there is still much room for improvements. In particular, towards increasing the watermarking payload while maintaining the simultaneous robustness against MP3 compression and additive noise attacks among other attacks.

3. COMPRESSED SENSING RELATED THEORY

Compressed sensing relies on the concept of a sparse domain representation of compressible signals. In the basic formulation by Candès and others [16-18], if a K-sparse vector x with dimension (N×1) is sampled with a random orthonormal Gaussian M×N matrix Φ producing a measurement vector y with dimension (M×1) where $M < N$, then, given the measurement vector and knowing the sampling matrix Φ it is required to recover the sparse vector x from y, where e is a small additive noise (in the general noisy case), and ε is its variance:

$$y = \Phi x + e \qquad (1)$$

An exact recovery of the sparse vector x is possible through L1-minimization using the basis pursuit denoising algorithm (BPDN) [16-19]. This is done by solving the convex optimization linear programming (LP) problem:

minimize $\|\boldsymbol{x}\|_{\boldsymbol{l1}}$

subject to $\|\boldsymbol{y} - \boldsymbol{\Phi x}\|_{l2} \leq \varepsilon$ (2)

If Φ satisfies the restricted isometry property (RIP), which is met for Gaussian random orthonormalized matrices [16] where columns are orthogonal, then recovery is possible under the following sparsity condition:

$$M > O\left(K\,Log\left(\frac{N}{K}\right)\right) \qquad (3)$$

Where *K* is the number of non-zero elements in *x*. More recently, there have been two extensions to this framework. The first by Laska et. al. [20] in what was named the "Justice Pursuit" (JP) algorithm explained as follows: If the measurement vector is corrupted by an interference that is sparse in some domain, equation (1) becomes:

$$y = \Phi x + \Omega\beta + e \qquad (4)$$

Where Ω is some full or partial transform or random matrix with orthonormal columns and dimensions (*M*×*L*) where $L \leq M$ and β is a sparse vector with possibly large amplitude non-zero components of length L and sparsity *k*. The authors have shown that both sparse vectors (x and β) can be recovered if the two matrices are incoherent and the following sparsity condition is satisfied:

$$M > O\left((K+k)\,Log\left(\frac{N+L}{K+k}\right)\right) \qquad (5)$$

Where *k* is the sparsity of β and *e* is a small additive noise.

Another interesting technique by Nguyen et. al. [21] called the "*extended LASSO"* assumes that the sparse vector β is additive directly with the measurement vector (thus of same dimension *M*). In that sense, it can be considered an extension of the JP algorithm with the matrix Ω being the identity matrix of size *M*×*M*. In both cases, the recovery algorithm assumes a new sparse vector U=[x β] of size *(N*+*L*)×1 and a new sensing matrix ψ=[Φ Ω] with size *M*×*(N*+*L)*, which is assumed to still satisfy the RIP condition, and the basis pursuit denoising becomes:

Minimize $\|\boldsymbol{U}\|_{l1}$
Subject to: $\|\boldsymbol{y} - \boldsymbol{\psi U}\|_{l2} \leq \varepsilon$ (6)

When the sparse vector U is estimated, its first *N* elements are those of *x* and the remaining L elements are those of β [20-21]. Many fast algorithms have been developed in the literature for solving the basis pursuit and the *LASSO* formulations, of those, the L1-magic library [22] was used for all the L1-minimization algorithms in the proposed technique in this paper. The L1-magic library solves the LP convex problem using the primal-dual algorithm [22].

4. SPARSE WATERMARKING PROPOSED FORMULATION

4.1 Sparse Watermark

The watermark is a sparse vector β of length L and with k non-zero components which take the binary values ±1 based on the required watermark value. In this paper, k=1 (only one non-zero element). For embedding in a measurement vector of length M, a random vector watermark signal W is generated:

$$W = \Omega\, \beta \qquad (7)$$

Where Ω is an orthogonalized iid random Gaussian matrix of size (M×L). In this paper $L < M$ and Ω does a random expansion of the sparse vector β similar to the spread spectrum spreading concept, however, with real random values instead of the binary values used in the SS. It is noted here that this proposed random real-valued watermark is an obvious change in paradigm from the traditional watermarking literature where the watermark vector is a random binary vector.

4.2 Sparse Host Signal

Audio (music and speech) signals are highly compressible signals for which sparse domains exist, and that what makes audio compression feasible. In this paper, the Karhunen-Loeve transform (KLT), the discrete cosine transform (DCT) and the Walsh-Hadamard transform (WHT) [14] are all used to find a sparse representation for the host audio signal. The KLT is a data-dependent transform (with adaptive transform matrix) which is an optimal transform since it produces the sparsest possible representation provided that the statistical properties don’t change. Since this work is concerned with MP3 music robust watermarking, a KLT matrix is learned for every song. The song audio signal is divided into frames of length M each, and the KLT matrix of dimension (M×M) is learned using 60 seconds of the song as training data. For MP3 sampling rate of 44,100 samples per second, this corresponds to (2646000/M) frames. Once the KLT matrix is learned, it is applied to each frame to produce the sparse domain coefficients.

Both the DCT and the WHT on the other hand are non-adaptive, fixed matrix transforms. The DCT have orthonormal matrix, where in this paper the WHT (M×M) matrix and its inverse are normalized by the root of M.

A sparsity of 50% is forced for all the audio frames by maintaining the highest M/2 coefficients and putting the rest to zero. This process produced sparsified audio signal average quality of 40dB for the KLT and 35dB for the DCT and WHT respectively with no perceptible quality loss. Experimental trials showed that this sparsity level strikes a good balance between maintaining the sparse signal quality and obtaining sparse-signal recovery. Many more different approaches may be taken, for example, taking all the coefficients which fall within a certain ratio relative to the largest coefficient, or making the coefficients approximately sparse by scaling them down.

Let the audio frame in the sparse domain be $\boldsymbol{x}$ which is now K-sparse (where K is M/2), then, going back to the time-domain we take the inverse of the sparsifying transform (KLT, DCT or WHT). The sparsified host signal in time domain is thus:

$$X(t) = Tx \qquad (8)$$

Where T is the inverse matrix of KLT, DCT or WHT.

4.3 Watermark Embedding

The watermarked host signal is given by:

$$y = Tx + \alpha.\Omega\,\beta \tag{9}$$

Where α is an embedding strength scalar factor which is adaptively adjusted to make the SNR of the watermarked signal constant at 28dB as follows. Since all watermark vectors (the columns of the Ω matrix) are normalized, then the SNR is given by:

$$SNR = 10Log_{10}\frac{\sum_{i=1}^{M} Xi^2}{\alpha^2} \tag{10}$$

Thus, fixing the SNR at 28dB, the embedding strength is adapted each frame using the formula:

$$\alpha = 0.04\sqrt{\sum_{i=1}^{M} Xi^2} \tag{11}$$

The adaptive value of α imposes no problem in the recovery, since it will only scale the sparse vector β but will not affect its sign which represents the watermark value. It is to be noted that if the location of the non-zero element in β is fixed, then one bit is embedded at each M-length frame since the sparse watermark contains one non-zero value. If the position of the non-zero element is allowed to change, then more bits are encoded in each frame since the location of the non-zero element combined with its sign give more information. A block diagram of the embedding process is shown in Fig.1, while examples of long and short segments of the music signal are shown in Fig.2 (left and right respectively).

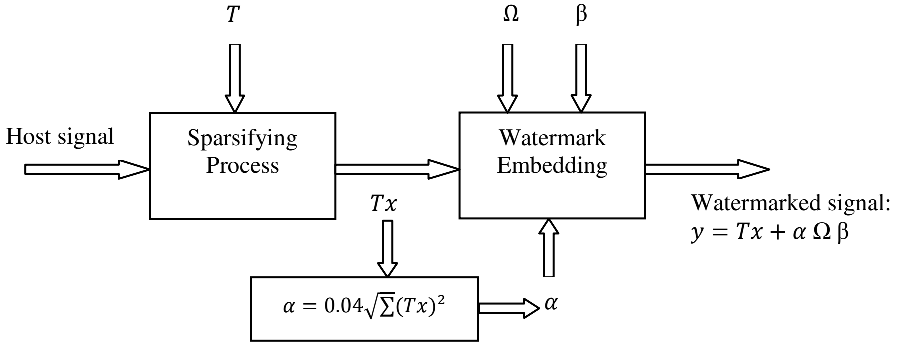

Figure 1. Diagram of the Watermark Embedding Process

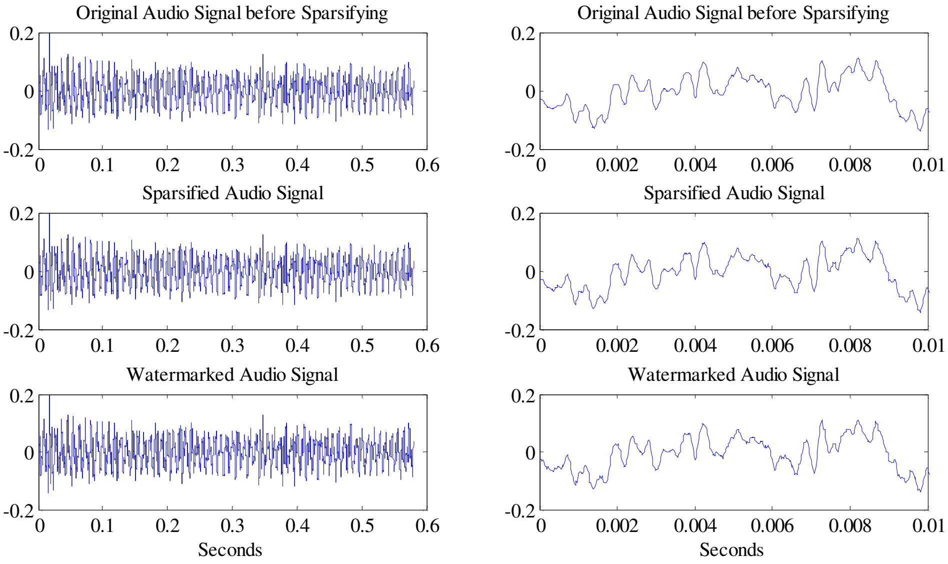

Figure 2. Original, sparsified and watermarked segments. Left: 0.6 sec. segment. Right: 0.01 sec. segment.

4.4 Host and Watermark Recovery

One major advantage of the proposed approach is that it can recover the host signal perfectly in clean conditions, an advantage which does not exist in most watermarking methods in the literature. This may have some important practical applications, for example, to use a perceptible watermark where the noisy host signal is free for previewing and downloading, but the clean signal can only be obtained by the L1-minimization recovery, which requires the secret random matrix Ω as shown in Fig. 3.

Three different, almost equivalent, methods are used in this paper to recover the sparse watermark and the host signal:

1) Direct Justice Pursuit:

Having the measurement vector *y*, we apply the basis pursuit denoising (BPDN) algorithm as in (6):

$$\textbf{\textit{Minimize}}\ \|\boldsymbol{U}\|_{l1}$$
$$\textbf{\textit{Subject to:}}\ \ \|\boldsymbol{y}-\ \boldsymbol{\psi U}\|_{l2} \leq \varepsilon$$

Where in this case $\boldsymbol{U}$= [x β] and $\boldsymbol{\psi}$ =[T Ω] with size $M \times (M+L)$. It is noted here that the values of M, L, K and k used in this paper satisfy the condition in equation (5) (and note that the N in (5) equals M in this paper since we use a square transform matrix) thus the condition in (5) becomes:

$$M > O\left((K+k)\ Log\left(\frac{M+L}{K+k}\right)\ \right) \qquad (12)$$

And the typical values used in this paper are (*M*=128 or 64, *L*=8, *K*=64 and *k*=1). The watermark sparse vector which contains only one non-zero value is β. With the sparse vector *x* is recovered, the recovered host signal in time domain is obtained by *X(t)=T x*. In the clean situation with no attacks, both the watermark and the host signal are recovered perfectly (with over 90dB for the host signal relative to the un-watermarked host). However, in non-ideal conditions, more than one non-zero value may appear in the recovered watermark. In this paper, we find the largest value and take its sign as the estimated watermark.

2) Multiplying by the Inverse of Ω*:* (Ω_I)

In this case, we multiply the watermarked host y by the inverse of the random matrix Ω (using Moore-Penrose pseudo inverse function in Matlab [15]) to have a new measurement vector y1=$\Omega_I y$ with a new dimension L and is given by:

$$y1 = \Omega_I Tx + \beta \qquad (13)$$

Following the basis pursuit denoising formulation in (6), we use $\boldsymbol{\psi}$ =[Ω_IT *I*] where *I* is the identity matrix of size L×L and we get the recovered sparse vector $\boldsymbol{U}$= [x β].

3)*Multiplying by the annihilator of* Ω: (Ω_{AN})

The annihilator of Ω is a matrix Ω_{AN} of dimension *(M-L)*×M. In this case, the new measurement vector y_{AN}=$\Omega_{AN} y$ with a new dimension *(M-L)* and is given by:

$$y_{AN} = \Omega_{AN}\ T\ x \qquad (14)$$

We use $\boldsymbol{\psi}$ =[Ω_{AN} T] and in this case, we only get the sparse vector *x*, and by getting *x* we do the subtraction $(y - Tx)$ which produces

$$y2 = \Omega\beta \qquad (15)$$

Then we can get β either multiplying by Ω_I or by applying the basis pursuit to get the sparse vector β .

The watermark value detection looks at the sign of the recovered watermark value. A simple voting is used between the three methods described to make the final decision. Experimental results have shown they are highly equivalent.

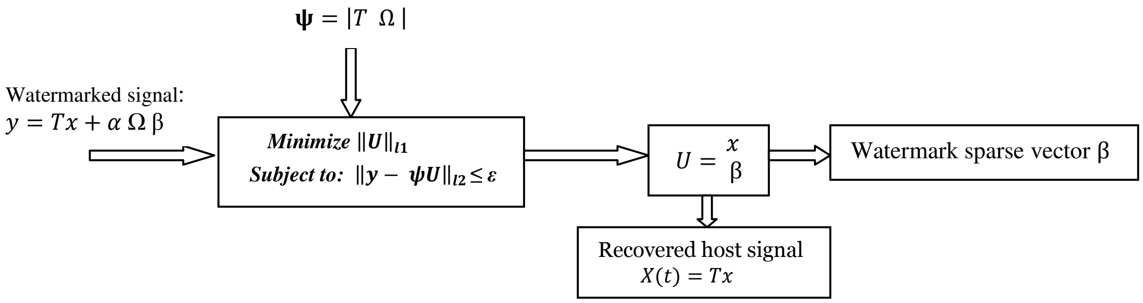
$\psi = |T\ \Omega|$
Watermarked signal:
$y = Tx + \alpha\,\Omega\,\beta$
Minimize $\|U\|_{l1}$
Subject to: $\|y - \psi U\|_{l2} \leq \varepsilon$
$U = \begin{matrix} x \\ \beta \end{matrix}$
Watermark sparse vector β
Recovered host signal
$X(t) = Tx$

Figure 3. Diagram of the Watermark Recovery Process

4.5 Averaging to Enhance Robustness

To overcome the effects of additive noise and compression/decompression attacks on watermark detection, the averaging option was adopted. The watermark information estimated from *D* frames is averaged and the watermark bit value is re-estimated based on the averaged information. For recovery method-1, the averaging is done over *y* vector for *D* frames and the BPDN is applied to recover the watermark from the averaged vector. For method-2, the averaging is done over $y1$ for *D* frames and for method-3 is done over $y2$. The value *D* corresponds to a redundancy coding of the watermark bits, in the sense that the same watermark bit is repeated *D* times over *D* successive frames, as a mean of channel coding to enhance the robustness. This of course results in a decreased payload as will be discussed in the experimental results section.

4.6 Expected Error in Watermark Recovery

Since the three recovery methods are almost equivalent, let us look at method-3 in more details. The recovered sparse vector of the host is practically $\tilde{x}$ where the L2-norm of the difference is:

$$\|b\|_{l2} = \|x - \tilde{x}\|_{l2} \qquad (16)$$

Its upper bound is given in [9] by the formula:

$$\|b\|_{l2} \leq C\sqrt{K}\,\frac{\lambda.M}{L} + \|e\|_{l2} \qquad (17)$$

Where *e* is additive noise (may be attributed to the attack in this case), *C* and λ are factors depending on the matrices Ω and *T* and the sparsity *K*. Following (15) and taking the multiplying by Ω_I approach, we get:

$$\Omega_I.y2 = \beta + \Omega_I.b \qquad (18)$$

Where the term $\Omega_I.b$ in (18) represents the error in the watermark sparse vector estimate. Since this sparse vector β contains only one non-zero element, and assuming we know its location (the j_{th} element), then we can deduce that the error term in (18) is given by:

$$E = \Omega_I.b = \sum_{i=1}^{M} \Omega_{Iji}\ b_i \qquad (19)$$

And since Ω_{Ij} is a random sequence with zero-mean and unity variance (A columns in the orthonormal random Gaussian matrix Ω_I), then by the central limit theorem, the expected value of *E* approaches zero with convergence speed with the order of $\frac{1}{M}$ and its variance is $\|b\|_{l2}$. When the averaging over *D* frames is used, the error variance is decreased by an order of *D* and the expected value of E approaches zero with convergence speed of $\frac{1}{M*D}$. Thus, better estimates of the watermark are obtained with larger *M* and with more averaged frames, however, with a decreased payload which equals $\frac{Fs}{M*D}$ where Fs is the sampling frequency of the MP3 audio and $M*D$ represents the number of host samples carrying a single watermark bit.

4.7 Frequency Domain Embedding

The proposed embedding technique is extended to frequency domain (and in principle, to any linearly transformed domain). Assuming that the host signal *X(t)* is transformed to another domain by a transform matrix *G,* equation (9) becomes:

$$\boldsymbol{y}^{\mathrm{t}} = G.Tx + \alpha.\Omega\beta \qquad (20)$$

Where $\boldsymbol{y^t}$ is the watermarked host in the transform domain. In this paper, G is taken as the discrete cosine transform (DCT) matrix of size (M×M). The recovery is done exactly the same way as in the time-domain embedding, however, instead of using the matrix T we use the matrix product $G * T$. Thus, in recovery we solve the following basis pursuit denoising problem:

$$\textbf{\textit{Minimize}}\ \|\boldsymbol{U}\|_{l1}$$
$$\textbf{\textit{Subject to:}}\ \|\boldsymbol{y^t} - \boldsymbol{\Psi U}\|_{l2} \leq \varepsilon$$

Where in this case $\boldsymbol{U}$= [x β] and $\boldsymbol{\Psi}$=[$G * T$ Ω]. If the matrix Ω used in this case is the same as the time-domain embedding then we get the same results since the frequency characteristics were not exploited. Instead, in this paper the matrix Ω is modified to make different embedding weight for different frequencies. As a preliminary experiment, the bandwidth of the host is divided into 3 parts (low, medium and high frequency bands). Thus, the dimension M of the matrix Ω is divided to 3 parts accordingly and multiplied by [0.4 0.9 0.1] respectively for the low, medium and high bands (these values are chosen experimentally). This puts more embedding strength in the mid-frequency band than the low and high frequency bands, since the low frequency band affects the quality and the high frequency band is most affected by MP3 and noise adding attacks.

4.8 Spread Spectrum (SS) as a Baseline for Comparison

Spread spectrum based watermarking is still a popular and reliable embedding technique for its robustness to additive noise [1]. It is taken as a baseline for comparison in this paper. The following formulation of the SS embedding and detection is intended to show the resemblance and differences with the CS approach. The embedding equation is given by:

$$y_{ss} = T.x + \mu.P\,W_{ss} \qquad (21)$$

Where $T.x$ is the sparsified signal in time domain, μ is an embedding factor which is taken proportional to the frame energy. P is a random matrix with ±1 binary values for spreading, and W_{ss} is a sparse vector of length M with only one non-zero element taking the +1 or -1 value representing the watermark bit value for this frame. The embedding factor μ is selected so that SS embedding would have the same SNR as the proposed compressed sensing (CS) embedding. The location of the non-zero value selects which column of the random binary matrix P would be used for embedding. In traditional SS this location is fixed a priori and known to both encoder and decoder. At recovery, the received watermarked measurement y_{ss} is multiplied by the corresponding column from the matrix P and the resulting sequence is summed and its sign is taken as the detected watermark value.

4.9 Related Work in CS Watermarking

In 2007 Shiekh and Baraniuk (SB technique for brevity) [25] proposed a transform domain image watermarking model based on compressed sensing as follows. Let $y_t = Af + e$ be the transform domain watermarked signal where f is the spread spectrum watermark sequence, A is an $m \times n$ random matrix where $(m>n)$ and e is the sparse transform domain vector for the host signal.

The annihilator of A, (A_{an}) is multiplied by the transform domain vector to give a new vector $y^{'} = A_{an}.e$ for which the L1-minimization is performed. Once the sparse transform domain signal e is detected, it is subtracted from y_t and the result is multiplied by the inverse of A to get the watermark f. One major difference between their technique and the one proposed here is that in this paper here the watermark itself is a sparse vector allowing for simultaneous recovery of the watermark and the host signal, and potentially better robustness against attacks by adaptively selecting the location of the non-zero element in the sparse watermark vector, which also allows us to encode more than one bit in each frame by combining the location with the sign. Furthermore, one can theoretically insert more than one non-zero element, since the L1-minimization can detect the positions and signs of the non-zero elements. In [25] the authors applied their technique to a fragile image watermarking scenario where no attacks were considered, and the main aim was to show that the watermark can be detected perfectly with zero-errors as long as the CS conditions are met. In this paper the (SB) technique is implemented in the context of audio watermarking and is compared to the proposed technique as shown in section 5.

5. EXPERIMENTAL RESULTS

5.1 Experimental Setup

To demonstrate the practicality of the proposed compressed sensing based watermarking technique, it was applied on a 128kbps MP3 music file of duration 120 seconds containing slow rock music with vocals. The MP3 music file is first converted into WAV format for the watermark embedding stage. The file is divided into frames of length M each (128 and 64 samples are used). The watermark embedding is done for each frame and the results show the average performance over the 41,344 frames used where Fs=44,100 Hz.

The embedding random matrix was regenerated every 32 frames. When additive noise is added it is done before the MP3 attack. For the MP3 compression attack, the watermarked WAV file is converted back to MP3, then, converted back to WAV format to apply the watermark detection process. Both 128kbps and 64kbps MP3 compression attacks were considered.

The watermark recovery performance is measured at 3 different points. First, after the watermark embedding and before writing back the MP3 file to measure the effects of additive noise alone. Secondly, after writing the MP3 compression-decompression, and finally, after applying the averaging process for both the noise attack and the MP3 attack. The watermark embedding strength in all experiments was fixed at an SNR of 28dB which gives minor effects to the host signal quality. The lower dimension of the CS embedding matrix Ω is taken L=8 with sparsity k=1 where the non-zero element position is fixed and known to the decoder, and only the sign is estimated. It is to be noted that in all experiments, the position of the non-zero element in the sparse watermark β is assumed to be known to the receiver. This position can be fixed all along the signal, or is allowed to change in a pre-determined fashion. In all the experiments, the position (between 1 and 8) is changed every 4 frames and is assumed to be known at the watermark extraction where the sign of the estimated sparse vector is checked only at that position. It is to be noted that the sparse watermark vector length L was varied in one 128kbps MP3 attack experiment from 2 to 40 with no obvious change in performance. However, in general, the larger this length becomes, the more likelihood that errors may occur in the position of the non-zero element since L column vectors in the Ω matrix compete, and thus, it is fixed to 8 in all the results presented in this paper.

The first experiment is the additive noise attack effects, where the SS and SB techniques are also included with and without averaging. The second experiment is the MP3 compression / decompression. In all experiments, the proposed technique is termed (CST) and (CSF) for time domain and frequency domain embedding respectively. The spread spectrum and the Shiekh-Baraniuk are termed (SS) and (SB).

5.2 Additive Noise Attack

Additive noise was added with increasing levels starting from the clean condition (28dB) until the SNR reached (8.5dB) and the noise was quite annoying. Note that starting from a SNR below 20dB the quality is not acceptable. Figure 4 shows the %success rate of watermark bits recovery for the proposed CST and CSF techniques as well as for the SS and SB techniques all with and without averaging. The left figure shows the no averaging case, corresponding to a payload of 344bps. The 3rd right figure with averaging factor D equals 2 corresponding to a payload of *172bps*. The results of this experiment give the following indications:

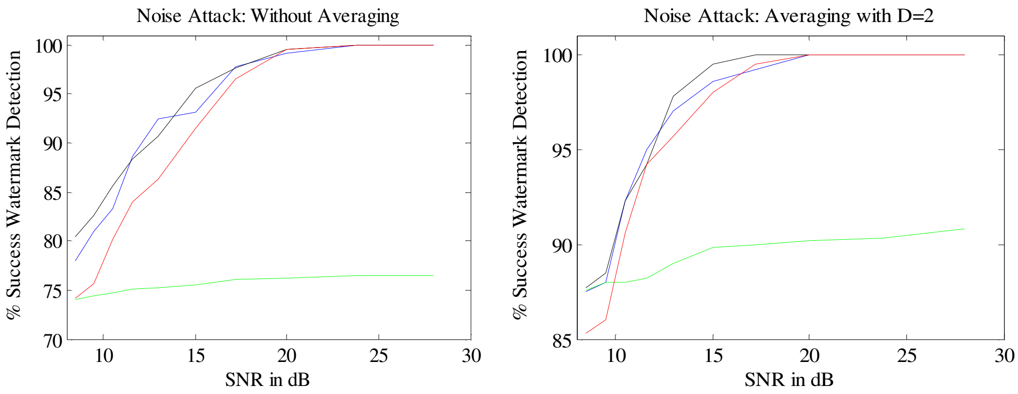

Figure 4. Noise attack performance comparison: **green**: SS, **blue**: CST, **black**: CSF, **red**: SB

1- The best performance is obtained by the proposed technique with frequency domain embedding and the spectral shaping. However, the advantage over the time domain embedding and the SB technique is not significant.

2- For a noise attack with 20dB (considered very noisy) the proposed watermark with averaging is highly robust with almost zero bit error rates.

3- Due to the host-rejection problem in the SS technique, an overhead of performance loss is always there as clear from the results in Figure 4.

5.2.1 Robustness to sever additive noise attack:

As seen in Table 2, it takes an averaging of 16 frames to reach acceptable bit error rate for the SS technique under sever additive noise attack with 10dB SNR. The table also shows a comparison between the performances of the competing techniques versus the number of averaged frames *D*, where the proposed techniques reach the same acceptable performance of 1% BER with *D*=4.

Table 2. *%Success Rate at* 10 dB as a function of *D*

D	*Payload rate (bps)*	*CST*	*CSF*	*SS*	*SB*
1	*344bps*	*83.3*	*85.7*	*74.7*	*80.2*
2	*172bps*	*92.3*	*92.3*	*88*	*90.6*
4	*86bps*	*99*	*99*	*90.7*	*98*
8	*43bps*	*100*	*100*	*95.6*	*100*
16	*22bps*	*100*	*100*	*99*	*100*

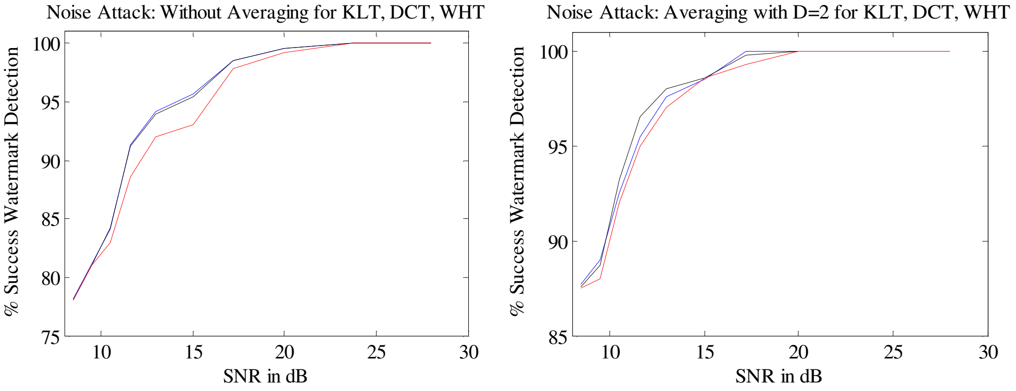

Figure 5. Noise attack performance comparison: **blue**: KLT, **black**: DCT, **red**: WHT

Figure 5 above shows a comparison between the different sparsifying transforms. In theory, the KLT should perform the best since it is the "optimal" transform, however, it is trained on only 60 seconds from the song, which contains large statistical variations, and thus in practice, and under the conditions used, it is not performing better than the DCT. The best performance is obtained by both the KLT and the DCT with slightly better performance than the WHT. It is noted however that the KLT is the best of the three in the sparsifying process quality since it produces an average of 40dB SNR compared to the other two which produce 35dB at 50% sparsity imposed on all transforms.

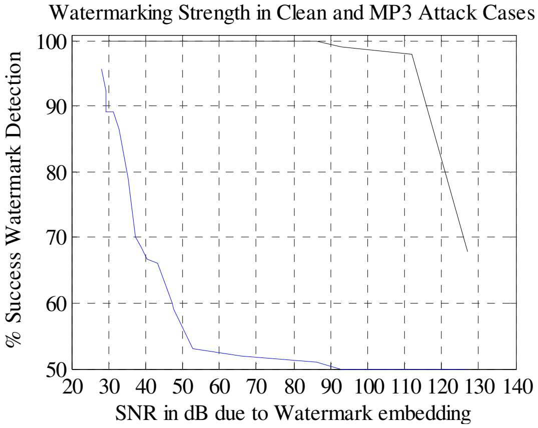

Figure 6. Effect of watermark embedding strength in clean and MP3 attack. **Blue**: MP3 attack. **Black**: No Attack

5.3 MP3 Compression/Decompression Attack

The MP3 compression/decompression attack is tested for the proposed technique. This attack is essential since in practice, one would take an MP3 music file, convert it into a WAV file, do the watermark embedding then convert back to the MP3. Hence, this conversion has to maintain a stable embedding and recovery for the watermark. Figure 6 above shows a comparison between the 128kbps MP3 attack versus the clean (no attack) cases as functions of watermark embedding strength. It is concluded that when the SNR goes below 28dB the MP3 attack performance degrades very quickly, and that is the reason why this watermark embedding strength is used in all the experiments. On the other hand, it is very interesting to see that the clean (WAV format) watermarked signal had preserved the watermark detection performance even with very weak watermark embedding strength (90dB). This may be a useful result in situations where the audio file is kept in the uncompressed (WAV format) and is used to hide important information in a fragile data hiding context.

Tables 3 and 4 show the %success rate of watermark bits recovery for the proposed CST and CSF techniques and for SS and SB techniques, with and without averaging with frame length M=128 for the 128kbps and the 64kbps MP3 compression/decompression attacks respectively. The first column shows the number of averaged frames D and the second column shows the corresponding payload. Results show that acceptable performance is obtained at payload rates of 43bps and 11bps for the 128kbps and 64kbps MP3 compression attacks with averaging of 8 frames and 32 frames respectively. Even though these results are not the highest in the literature for MP3 attacks, as shown in table 5, when the additive noise robustness is also considered, it makes the proposed technique a reliable candidate since most reported high payload techniques in the literature are based on QIM approach which is very sensitive to additive noise attacks. Moreover, most of those techniques rely on hiding the watermark information in a pre-determined, specific band (frequency bands or wavelet details band), and thus are much easier to target in comparison with the proposed technique which spreads the watermark embedding over the whole frame length in time, and over the whole frequency band. Another advantage is that the proposed technique offers zero error recovery and potentially very large payload in clean conditions, which is useful in fragile data hiding situations.

Table 3. 128kbps MP3 attack on watermark %success rate (M=128)

D	***Payload rate (bps)***	***CST***	***CSF***	***SS***	***SB***
1	*344bps*	*95.8*	*97*	*62*	*95.6*
2	*172bps*	*97.4*	*98.5*	*67.5*	*97*
4	***86bps***	*98.4*	***99.3***	*78*	*98*
8	***43bps***	***99.6***	***99.5***	*91.6*	*99.2*
12	*29bps*	*100*	*100*	*96.9*	*100*
16	*22bps*	*100*	*100*	*98.2*	*100*

Table 4. 64kbps MP3 attack on watermark %success rate (M=128)

D	***Payload rate (bps)***	***CST***	***CSF***	***SS***	***SB***
1	*344bps*	*70.2*	*72*	*57*	*70.2*
4	*86bps*	*77.5*	*78*	*65.4*	*76.6*
8	*43bps*	*91*	*91.8*	*72.6*	*90.4*
12	*29bps*	*93.2*	*94*	*77.8*	*92*
16	*22bps*	*94.6*	*95*	*86*	*93.7*
32	***11bps***	*98.5*	***99.5***	*91.3*	*98.1*

Table 5. Proposed technique in comparison with Table 1

Algorithm	**MP3 Quality**	**SNR**	**%BER**	**Payload**
Proposed	64kbps	28dB	0.5	11 bps
Proposed	128kbps	28dB	0.7	86bps

5.4 Recovery of the Host Signal using L1-Norm Minimization

One of the major differences between the compressed sensing framework and the other embedding approaches the proposed technique can recover the host signal and separate it blindly from the watermark. The recovered host signal SNR is tested in 3 situations, in the clean condition case, after the additive noise and after the MP3 attack. In the clean condition, an SNR of more than *90dB* is obtained from the 28dB watermarked signal. This is quite remarkable and it shows that the original host signal can be almost perfectly recovered from the watermarked one. In the additive noise case, an average gain of +3dB is obtained over the noisy watermarked signal. In the MP3 attack, an average gain of +2dB is obtained over the MP3 compressed-decompressed signal. Such gains can be enhanced with a careful adjustment of the parameters of the basis pursuit denoising algorithm for the additive noise case, and by characterizing the nonlinear distortion caused by the MP3 attack.

6. Conclusions And Future Work

A new technique for watermarking “sparse watermarking” based on the compressed sensing framework is proposed. The sparse watermark vector which contains one non-zero element is expanded by a random matrix and added to the host signal, which is made sparse in a specific transform domain. The proposed technique relies on the compressed sensing framework and the L1-minimization for watermark and host signals recovery. The proposed technique is applied to audio watermarking and it works perfectly for clean conditions with no attacks. Additive noise and MP3 compression attacks are used to test the practicality of the proposed technique, in comparison with spread spectrum watermarking, with significant advantage for the proposed technique. The proposed technique has 3 main advantages over recent techniques in the literature. Firstly, the embedding is secure and distributed as opposed to specific coefficients embedding. Secondly, it is robust to additive noise as opposed to QIM based techniques, and finally, it can recover the host signal perfectly in clean conditions. More research needs to be done in the following issues. 1- Characterizing the MP3 distortion effects so that the L1-minimization can recover the host signal with better SNR under MP3 attacks. 2- Investigating the issue of using more non-zero elements in the watermark sparse vector and changing their positions so the decoder estimates both the sign and the position. 3- Investigating how to find optimal embedding matrix and optimal weighting for frequency domain embedding, and finally, 4- Optimally selecting a random watermark vector from the random matrix on a frame by frame basis to optimize robustness.

REFERENCES

[1] T. K. Tewari, V. Saxena, J. P. Gupta, “Audio Watermarking: Current State of Art and Future Objectives”. International Journal of Digital Content Technology and Applications, Vol. 5, No. 7, pp. 306-313 July 2011.

[2] R. M. Noriega, M. Nakano, B. Kurkoski and K. Yamaguchi, “High Payload Audio Watermarking: toward Channel Characterization of MP3 Compression”. Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 2, pp. 91-107 April 2011.

[3] Vivekananda Bhat, K•I. Sengupta, A. Das, “An Audio Watermarking Scheme using Singular Value Decomposition and Dither-Modulation Quantization”. Multimedia Tools and Applications Journal, Vol. 52, No. 2-3, pp. 369-383, 2011.

[4] S. Xiang “Histogram based Audio Watermarking against Time Scale Modification and Cropping Attacks”, IEEE Trans. On Multimedia, Vol. 9, Issue 7, pp. 1357-1372, November 2007.

[5] S. Xiang, "Robust Audio Watermarking by using Low Frequency Histogram". H.-J. Kim, Y. Shi, and M. Barni (Eds.): IWDW 2010, LNCS 6526, pp. 134–147, 2011.

[6] M. L. Dutta, Vinay Pathak, P. Gupta, "An Adaptive Robust Watermarking Algorithm for Audio Signals using SVD". M.L. Gavrilova. al. (Eds), Trans. On Comp. Sci. NLCS 6340, pp.131-153, 2010, Springer-Verlag.

[7] M. L. Dutta, Vinay Pathak, P. Gupta, "A Robust Watermarking Algorithm for Audio Signals using SVD". S. Ranka et. al. (Eds), IC3 2010, Part I, CCIS 94, pp. 84–93, 2010, Springer-Verlag.

[8] S. Dhavale, R.S. Deodhar, L.M. Patnaik, "Walsh Hadamard Transform Based Blind Watermarking for Digital Audio Copyright Protection". In V.V. Das and N. Thankachan (Eds.) CIIT 2011, CCIS 250, pp.469-475.

[9] S. Dhavale, R.S. Deodhar, L.M. Patnaik, "Lossless Audio Watermarking Based on the Alpha Statistic Modulation", International Journal of Multimedia & Its Applications (IJMA), Vol. 4, No. 4, August 2012, pp.109-119.

[10] H. Yang, D. Bao, X. Wang, P. Niu, "A Robust Content Based Audio Watermarking using UDWT and Invariant Histogram". Multimedia Tools and Applications Journal, Springer, On-line Nov. 2010.

[11] N. El Hamdouni, A. Adib, S. Labri, M. Torki, "A blind digital audio watermarking scheme based on EMD and UISA techniques". Multimedia Tools and Applications Journal, Springer, On-line, Jan. 2012.

[12] S. Wu, J. Huang, D. Huang, Y.Q. Shi, "Efficiently self-synchronized audio watermarking for assured audio data transmission", IEEE Trans. Broadcasting", vol. 51, no. 1, pp. 69-76, 2005.

[13] E. Ercelebi and L. Batakci, “Audio watermarking scheme based on embedding strategy in low frequency components with a binary image”. Digit. Signal Process., vol. 19, no. 2, pp. 265–277, 2009.

[14] R. Wang, D. Xu, J. Chen, C. Du, "Digital Audio Watermarking Algorithm based on Linear Predictive Coding in Wavelet Domain". Proc. of Int. Conf. on Signal Processing,vol. 3, pp. 2393-2396, 2004.

[15] R. Gunjan, P. Pandia, “Robust and Energy Efficient Audio Watermarking Scheme Resiliant to Desynchronization Attacks". V.V. Das and N. Thankachan (Eds.), CIIT 2011, CCIS250, pp.368-374, 2011, Springer-Verlag.

[16] Emmanuel Candès and Terence Tao, “Decoding by linear programming”. IEEE Trans. on Information Theory, 51(12), pp. 4203 - 4215, December 2005

[17] Emmanuel Candès and Terence Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” IEEE Trans. on Information Theory, 52(12), pp. 5406 - 5425, December 2006

[18] Emmanuel Candès and Paige Randall, “Highly robust error correction by convex programming”. IEEE Transactions on Information Theory (2006) Vol. 54, Issue: 7.

[19] Mark Davenport, Marco Duarte, Yonina Eldar, and Gitta Kutyniok, "Introduction to compressed sensing", (Chapter in Compressed Sensing: Theory and Applications, Cambridge University Press, 2012)

[20] Jason Laska, Mark Davenport, and Richard Baraniuk, Exact signal recovery from sparsely corrupted measurements through the pursuit of justice. (Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, California, November 2009)

[21] N. H. Nguyen, N. M. Nasrabadi, T. D. Tran, “Robust Lasso with Missing and Grossly Corrupted Observations”. NIPS 2011.

[22] L1-magic Library Website: http://users.ece.gatech.edu/~justin/l1magic/

[23] Roy and G. Saha, “Application of a Novel Audio Compression Scheme in Automatic Music Recommendation, Digital Rights Management and Audio Fingerprinting”. World Academy of Science, Engineering and Technology Journal, 48-2008, pp.649-659.

[24] Matlab 2012. http://www.mathworks.com/help/techdoc/ref/pinv.html

[25] Mona Sheikh and Richard Baraniuk, Blind error-free detection of transform-domain watermarks. (IEEE Int. Conf. on Image Processing (ICIP), San Antonio, Texas, September 2007) .