*Article*

# Compressive Sampling with Multiple Bits Spread Spectrum-Based Data Hiding

**Gelar Budiman**[1,†,‡*], **Andriyan B. Suksmono** [2,‡] **and Donny Danudirdjo** [3,‡]

[1] Affiliation 1; gelarbudiman@telkomuniversity.ac.id
[2] Affiliation 2; suksmono@stei.itb.ac.id
[3] Affiliation 3; donny@stei.itb.ac.id
* Correspondence: gelarbudiman@telkomuniversity.ac.id
† Current address: Telkom University Jl. Telekomunikasi Terusan Buahbatu Bandung
‡ These authors contributed equally to this work.

**Abstract:** We propose a novel data hiding method in an audio host with a compressive sampling technique. An over-complete dictionary represents a group of the watermark. Each row of the dictionary is a Hadamard sequence representing multiple bits of the watermark. Then, the singular values of segment-based host audio in a diagonal matrix multiply by the over-complete dictionary producing a lower size matrix. At the same time, we embed the watermark into the compressed audio. In the detector, we detect the watermark and reconstruct the audio. This proposed method offers not only hiding the information but also compressing the audio host. The application of the proposed method is a broadcast monitoring and biomedical signal recording. We can mark and secure the signal content by hiding the watermark inside the signal while we compress the signal for memory efficiency. We evaluate the performance in terms of payload, compression ratio, audio quality, and watermark quality. The proposed method can hide the data imperceptibly, in range 729-5292 bps with compression ratio 1.47-4.84 and perfect detected watermark.

**Keywords:** Compressive sampling, compressed sensing, watermark, data hiding, spread spectrum, singular value decomposition, Hadamard.

## 1. Introduction

At present, the exchange of data and information in the internet network has increased very dramatically. With more and more people accessing the internet and more and more content that can be accessed, the size of the data accessed in a given time increases on an exponential scale. With the increase in data access, more and more crimes related to data include data falsification, data theft, claiming unilateral ownership of data, leaking data, deception of data, and many other crimes related to internet data access. These problems have implications for the more losses experienced by data owners, which also affect state losses. Losses suffered by the state causes harm to its people. So that crime in the internet world only benefits certain parties and gives a big loss for the wider community. Thus technology that provides security for data, including marking ownership rights to data and hiding important data when sent over the internet, becomes mandatory to anticipate losses suffered by the wider community.

With the more and more data content accessed, the greater the memory capacity needed, besides assuming the network infrastructure does not increase, the network capacity also decreases due to increased data traffic accessed, and power requirements on the network infrastructure also increase. These conditions bring problems in how to access data efficiently so that we can save the infrastructure and energy needs to minimum usage. One technique that can provide solutions to these problems is

Compressive Sampling or Compressed Sensing (CS). This technique takes or picks up part of the data or signal from the sensor and then sends the data from the sample, and the receiver can reconstruct it back to the data as if it were authentic.

In this paper, we propose a technique for sampling audio signals and inserting or hiding data into them at the same time, so that the sampled signals have a smaller size and at the same time there is data inserted into the encoded data. With this technique, the signal stored in the cloud system from recording results by sampling is smaller in size, and we can mark it with hidden data at the same time. The broadcast monitoring application is an example of how to monitor signals in real-time and stores the results into the cloud. Monitoring such signals is more efficient if partial signal sampling is applied, such that the signal size becomes smaller than the original signal. At the same time, marking or indexing is applied by hiding data on the signal at any given duration to secure the authenticity of the monitored signal or to index the monitored signal by hiding its index on the encoded signal. Another application example is the recording of biomedical signals in which one sampled them using several sensors, and at the same time, embeds the ownership marking or index into the encoded signal. Thus the recorded biomedical signal has a smaller size than the original size but does not reduce the quality of the biomedical signal, and there is a marking that is inserted in the encoded signal to secure the biomedical signal.

CS in the audio combining with the data hiding technique is a rare topic. The combination of CS and data hiding makes it possible to compress the audio and, at the same time, to hide the watermark. Hua in [1] and Xin in [2] formerly proposed the CS applications in the audio combining with data hiding. In [2], Xin proposed an embedding method on host audio that is semi-fragile zero-watermarking by decomposing the audio in the wavelet domain and applying the CS technique to the audio wavelet coefficients without describing the audio reconstruction to determine the audio quality after the embedding process.

Xin on [2] proposes an insertion method on audio that is semi-fragile by first decomposing audio in the wavelet domain and applying the CS technique to the audio wavelet coefficient. Watermarks inserted in the measurement vector utilize positive and negative signs on the matrix elements. The result is that the inserted watermark is resistant to damage samples from the signal. But this paper does not explain the function of CS in terms of reducing signal size. Xin only explained CS techniques as a technique of inserting data with the properties of semi-fragile.

Griffin, in [3], proposed the CS method to compress the sinusoidal signal. Griffin investigated whether CS can be used to compress sinusoidal audio at a low speed because audio models like this have a high degree of spacing in the frequency domain. In his proposed method, Griffin performed CS techniques on single channels and multi-channel of audio signals with sinusoidal characteristics only. Griffin stated that the research he did was not to develop audio compression techniques and compare with existing compression techniques, but to find out how far CS was able to be applied in reducing the size of audio files so that the application applied, in this case, was for wireless sensor networks. Griffin can produce the smallest compression ratio by 5.4%. He applied spectral whitening first on the new audio then applied the CS technique to the spectral results, so that produced a tiny compression ratio with good quality of reconstruction results.

Fakhr on [4] proposed an insertion method using CS techniques by first thinning the host audio and watermark signals using Walsh Hadamard Transform (WHT), Discrete Cosine Transform (DCT) and Karhunen-Loeve Transform (TLC). Watermark extraction and audio host are done by reconstruction $L_1$ minimization. Fakhr claimed that the technique could withstand MP3 attacks at the lowest rate of 64 kbps with an 11 bps watermark payload, and the highest payload at 172 bps against additive noise attacks. But Fakhr uses CS not for compression techniques but as an insert technique. Fakhr uses MP3 attacks as compression to reduce the size of the audio signal after embedding a watermark.

In [1], Hua proposed a data hiding technique which combined with CS synthetically. Suppose we define an over-complete dictionary $\mathbf{A} \in \mathbb{R}^{p \times r}$, an uncompressed vector $\mathbf{z} \in \mathbb{R}^{r \times 1}$, a watermark bit to be inserted as $b \in \{-1, +1\}$, a watermark code sequence $\mathbf{w} \in \mathbb{R}^{r \times 1}$, a compressed vector $\mathbf{y} \in \mathbb{R}^{p \times 1}$ and $\alpha$ as gain control of the watermark, then we have

$$\mathbf{y} = \mathbf{A}(\mathbf{z} + \alpha b \mathbf{w}), \tag{1}$$

Hua inserted $b$ as the additional operation to $\mathbf{z}$ after multiplied by $\alpha\mathbf{w}$. In this paper, we embed the watermark bits into the over-complete matrix $\mathbf{A}$. Then we multiply $\mathbf{A}$ to the diagonal matrix from the singular values of host audio.

The data hiding technique proposed in this paper is multiple orthogonal codes on Spread Spectrum (SS) based as formerly introduced by Xin on [5] in time domain embedding and continued by Xiang in DCT domain embedding on [6] and [7]. We use the Hadamard code as the sequence for multiple bits of the watermark due to its best code performance [8]. The matrix $\mathbf{A}$ consists of $p$ Hadamard sequences that represent $p$ groups of multiple bits.

One of the signal sparsity techniques is a shrinkage technique on Singular Value Decomposition (SVD) output. This SVD technique truncates $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ with specific rank as also described in [9], [10], and [11]. This shrinkage technique yields a more compressed signal as the CS output, but certainly decreases the quality of the reconstructed signal. In this paper, we decompose a host signal using SVD. Then, the output of SVD, i.e., $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$, are truncated at a specific rank. We transformed the truncated singular matrix $\mathbf{S}_r$ to compressed domain $\mathbf{Y}$ via an over-complete dictionary containing SS-based data hiding $\mathbf{A}$. Thus, the ready matrices to be transmitted to the detector are $\mathbf{U}_r$, $\mathbf{Y}$, and $\mathbf{V}_r$. Then, in the receiver, firstly, we detect dictionary $\mathbf{A}$ containing the hidden data. We can extract the hidden data from the detected dictionary. Not only can we take back the hidden data, but also we can get the reconstructed signal to the original domain. Note that the process on the receiver needs only the compressed domain signal, such as $\mathbf{U}_r$, $\mathbf{Y}$, and $\mathbf{V}_r$. There is no dictionary and original data needs for data detection and signal reconstruction.

We organize the rest of this paper as follows. Section II describes the sparsity of singular value and CS technique for the audio compression. Section III explains the mathematics model and derivation of audio watermarking including the embedding, the extraction, the audio reconstruction process and the effect of the noisy environment to this proposed method. Section IV discusses the result of the simulation, while Section V reports the conclusion of this paper.

## 2. Sparse Singular Value and CS technique

The host signal in the form of a vector $\mathbf{x} = [x_1, x_2, \cdots x_L] \in \mathbb{R}^{1 \times L}$ is converted to a 2-dimensional matrix $\mathbf{X} \in \mathbb{R}^{M \times M}$ where $L = M^2$. The conversion to a 2-dimensional matrix X is applied in such a way that it produces

$$\mathbf{X} = \begin{bmatrix} x_1 & x_{M+1} & \cdots & x_{M(M-1)+1} \\ x_2 & x_{M+2} & \cdots & x_{M(M-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_M & x_{2M} & \cdots & x_{M^2} \end{bmatrix}. \tag{2}$$

The SVD process of $\mathbf{X}$ obtains orthogonal matrices $\mathbf{U} \in \mathbb{R}^{M \times M}$, $\mathbf{S} \in \mathbb{R}^{M \times M}$, and $\mathbf{V} \in \mathbb{R}^{M \times M}$, where the relationship is described as

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{3}$$

where $\mathbf{S}$ is a sparse diagonal matrix having $M$ non-zero elements in diagonal of the matrix as $M$ singular values. For compression needs, $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ can be truncated or reduced to $\mathbf{U_r} = \mathbf{U}[1, .., M; 1, .., r] \in \mathbb{R}^{M \times r}$, $\mathbf{S_r} = \mathbf{S}[1, .., r; 1, .., r] \in \mathbb{R}^{r \times r}$, and $\mathbf{V_r} = \mathbf{V}[1, .., M; 1, .., r] \in \mathbb{R}^{M \times r}$ with $r < M$. Then, we apply CS acquisition $\mathbf{S_r}$ as

$$\mathbf{Y} = \mathbf{A}\mathbf{S_r}, \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{p\times r}$ is an over-complete dictionary containing SS-based encoded watermark, $\mathbf{Y} \in \mathbb{R}^{p\times r}$ is an output of CS acquisition with smaller size than $\mathbf{S}$. The truncated matrix $\mathbf{S_r}$ has the form of

$$\mathbf{S_r} = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}, \tag{5}$$

where $\sigma_1, \sigma_2, ..., \sigma_r$ are the singular value elements. The matrix $\mathbf{A}$ is described later in Subsection 3.1. Finally, we have three matrices to be transmitted, that is $\mathbf{U_r}$, $\mathbf{V_r}$, and $\mathbf{Y}$. From this result, we can calculate the Compression Ratio (CR) as the comparison between the original signal length and the transmitted signal length as

$$\mathrm{CR} = \frac{L_X}{L_T} = \frac{M^2}{2Mr + pr}, \tag{6}$$

where $L_X$ is the elements number of $\mathbf{X}$, that is $M^2$ and $L_T$ is the total number of the transmitted elements $\mathbf{U_r}$, $\mathbf{Y}$, and $\mathbf{V_r}$, i.e. $2Mr + pr$.

We can calculate the reconstructed audio matrix with the same size as $\mathbf{X}$ in the form of

$$\mathbf{X_r} = \mathbf{U_r}\widehat{\mathbf{S}}_\mathbf{r}\mathbf{V_r}^T = \begin{bmatrix} \hat{x}_1 & \hat{x}_{M+1} & \cdots & \hat{x}_{M(M-1)+1} \\ \hat{x}_2 & \hat{x}_{M+2} & \cdots & \hat{x}_{M(M-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_M & \hat{x}_{2M} & \cdots & \hat{x}_{M^2} \end{bmatrix}, \tag{7}$$

where $\mathbf{X_r} \in \mathbb{R}^{M\times M}$ but its element values are slightly different than $\mathbf{X}$. The $r$ value controls the signal quality and the signal compression ratio. If $r$ is lower, then the compression ratio is higher but the signal quality is worse. Finally, we can get $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \cdots \hat{x}_{M^2}]$ as a reconstructed or decompressed version of the signal by converting back 2 dimensions matrix $\mathbf{X_r}$ to a vector or one dimensional signal $\hat{\mathbf{x}}$, thus we can calculate the signal quality by comparing $\mathbf{x}$ and $\hat{\mathbf{x}}$.

## 3. Data Hiding Model

### 3.1. An Overcomplete Dictionary with SS-based Content

In this proposed method, firstly, we convert the audio host to the frequency domain using DCT in the process before applying insertion and compression. In the audio receiver, after being reconstructed or decompressed, the reconstructed audio is re-converted to the time domain with IDCT. The DCT and Inverse DCT (IDCT) formulations used for this method are in the following equation [12]

$$X(k) = w(k) \sum_{n=0}^{N_p-1} x(n) \cos\left(\frac{\pi(2n-1)(k-1)}{2N_p}\right) \tag{8}$$

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

where $X(k)$ is the audio signal in the DCT domain, $x(n)$ is the audio signal in the time domain, and $N_p$ is the number of DCT points. While $l(k)$ is defined in the following equation

$$l(k) = \begin{cases} \frac{1}{\sqrt{N}}, \text{if } k = 0 \\ \frac{2}{\sqrt{N}}, \text{if } 1 \le k \le N_p - 1 \end{cases}. \tag{10}$$

In this paper, the orthogonal code mapping to multiple bit watermarks is a Hadamard sequence which taken from the Hadamard matrix. Denote the Hadamard matrix $\mathbf{H}_r \in \{-1, +1\}^{r \times r}$ generated by [13,14] as

$$\mathbf{H}_r = \begin{bmatrix} \mathbf{H}_{r\backslash 2} & \mathbf{H}_{r\backslash 2} \\ \mathbf{H}_{r\backslash 2} & -\mathbf{H}_{r\backslash 2} \end{bmatrix}, \tag{11}$$

where $\mathbf{H}_1 = [1]$. Assume $\mathbf{H}_r(j)$ is a vector from $j$-th row of $\mathbf{H}_r$, then the orthogonal Hadamard sequence $\mathbf{p}_j$, where $j = 1, 2, ..., r$, are obtained from

$$\mathbf{p}_j = \mathbf{H}_r(j). \tag{12}$$

Let $\mathbf{A}_0 \in \{-1, +1\}^{p \times r}$ be an SS-based content matrix, where $p < r$ and $\mathbf{p}_{t_i} \in \mathbb{R}^{1 \times r}$ is a Hadamard sequence associated with the watermark bits in $i$-th row of $\mathbf{A}_0$, and $t_i = \{t_1, t_2, ..., t_p\}$ be the set of Hadamard sequence indices where $i$ is a row index of $\mathbf{A}_0$. Thus $\mathbf{A}_0$ contains $\mathbf{p}_{t_i}$ as

$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{p}_{t_1} \\ \mathbf{p}_{t_2} \\ \vdots \\ \mathbf{p}_{t_p} \end{bmatrix}, \tag{13}$$

where semicolon from (13) restrict each $\mathbf{p}_{t_i}$ to the different row. Since there are $p$ rows of $\mathbf{A}_0$, there are $p$ Hadamard sequences in $\mathbf{A}_0$. Thus, we have an over-complete dictionary $\mathbf{A} \in \mathbb{R}^{p \times r}$

$$\mathbf{A} = \frac{1}{p}\mathbf{A}_0, \tag{14}$$

with unit norm of its columns: $\|a_m\|_2^2 = 1$, where $m = 1, 2, ..., r$.

A Hadamard sequence represents multiple watermark bits. Assume that there are $N_s$ watermark bits for a Hadamard sequence, then there are $N_p$ different Hadamard sequence possibilities, where $N_p = 2^{N_s}$. Note that the length of a Hadamard sequence and also the row of matrix $\mathbf{A}$ is $r$ bits, thus $r = N_p$ due to the square size of Hadamard matrix as (11). Denote $\mathbf{w}_{t_i}$ as a watermark vector in $i$-th segment of the watermark with a vector index or a Hadamard index $t_i$, then

$$\mathbf{w}_{t_i} = [w_{t_i}(1) \quad w_{t_i}(2) \quad \cdots \quad w_{t_i}(N_s)], \tag{15}$$

where $w_{t_i}(l) \in \{-1, +1\}$ and $l = 1, 2, ..., N_s$. In multi bit SS, the watermark vector $\mathbf{w}_{t_i}$ is mapped to a Hadamard sequence $\mathbf{p}_{t_i}$. For example, if we have 3 bits watermark in a Hadamard sequence, or $N_s = 3$ bits, then $N_p = 2^{N_s} = 8$ bits, thus all watermark possibilities and their mapping to Hadamard sequences are displayed in Table 1. If we have 2 segments or 2 vectors of watermark $\mathbf{w}_{t_1} = [-1, +1, -1]$ and $\mathbf{w}_{t_2} = [+1, -1, -1]$, then using Table 1 we get $t_1 = 3$ and $t_2 = 5$, thus $\mathbf{p}_{t_1} = \mathbf{p}_3 = \{+1, +1, -1, -1, +1, +1, -1, -1\}$ and $\mathbf{p}_{t_2} = \mathbf{p}_5 = \{+1, +1, +1, +1, -1, -1, -1, -1\}$.

The over-complete matrix $\mathbf{A}_0$ contains $pN_s$ bits of watermark for the host with length $M^2$ thus we can compute watermark payload $C$ in bps as

$$C = \frac{pN_sF_s}{M^2}, \tag{16}$$

where $F_s$ is the host signal sampling rate in sample/s. Due to $N_s = \lceil \log_2 N_p \rceil = \lceil \log_2 r \rceil$, thus (16) will be as

$$C = \frac{p\lceil \log_2 r \rceil F_s}{M^2}. \tag{17}$$

Once $\mathbf{A}$ is generated from the associated watermark bits, it is embedded into $\mathbf{S_r}$ using matrix multiplication in (4). The result $\mathbf{Y}$ is not only a matrix with a smaller size than $\mathbf{S_r}$, but also it is

**Table 1.** Watermarks and Hadamard sequences Example for $N_s = 3$, $N_p = 8$ and $r = 8$

| Index ($t_i$) | Watermark Bits ($\mathbf{w}_i$) | Hadamard Sequence ($\mathbf{p}_{t_i}$) |
|---|---|---|
| 1 | {-1,-1,-1} | {+1,+1,+1,+1,+1,+1,+1,+1} |
| 2 | {-1,-1,+1} | {+1,-1,+1,-1,+1,-1,+1,-1} |
| 3 | {-1,+1,-1} | {+1,+1,-1,-1,+1,+1,-1,-1} |
| 4 | {-1,+1,+1} | {+1,-1,-1,+1,+1,-1,-1,+1} |
| 5 | {+1,-1,-1} | {+1,+1,+1,+1,-1,-1,-1,-1} |
| 6 | {+1,-1,+1} | {+1,-1,+1,-1,-1,+1,-1,+1} |
| 7 | {+1,+1,-1} | {+1,+1,-1,-1,-1,-1,+1,+1} |
| 8 | {+1,+1,+1} | {+1,-1,-1,+1,-1,+1,+1,-1} |

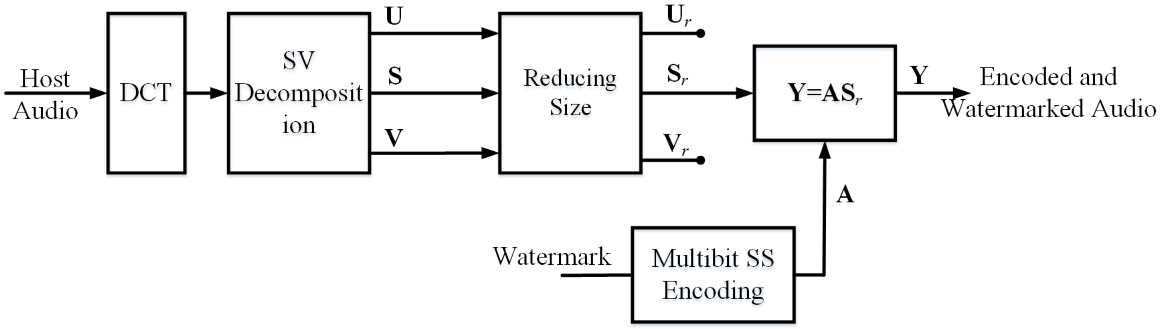


**Figure 1.** Watermark Embedding and Audio Encoding

embedded by the watermark bits. The matrix $\mathbf{S_r}$ is a diagonal matrix whose size is reduced from the original one $\mathbf{S}$. From (4), (5) and (13), the equation $\mathbf{Y} = \mathbf{AS_r}$ can be exploited as

$$\begin{bmatrix} \mathbf{y}_{t_1} \\ \mathbf{y}_{t_2} \\ \vdots \\ \mathbf{y}_{t_p} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{t_1} \\ \mathbf{p}_{t_2} \\ \vdots \\ \mathbf{p}_{t_p} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix}, \tag{18}$$

where $\mathbf{y}_{t_i} \in \mathbb{R}^{1\times r}$ is a vector of matrix $\mathbf{Y}$ at row $i$ which also corresponds to $\mathbf{p}_{t_i}$ and $\sigma_1, \sigma_2, ..., \sigma_r$ are singular value elements of $\mathbf{S_r}$. Each row of $\mathbf{A}$ or $\mathbf{p}_{t_i}$ is a vector with size $1 \times r$. $\mathbf{S_r}$ is a diagonal matrix with size $r \times r$. Thus, we can simplify (18) to the following equation

$$\begin{bmatrix} \mathbf{y}_{t_1} \\ \mathbf{y}_{t_2} \\ \vdots \\ \mathbf{y}_{t_p} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{t_1}\mathbf{S_r} \\ \mathbf{p}_{t_2}\mathbf{S_r} \\ \vdots \\ \mathbf{p}_{t_p}\mathbf{S_r} \end{bmatrix} \tag{19}$$

Then, we can have the following simple vector expression

$$\mathbf{y}_{t_i} = \mathbf{p}_{t_i}\mathbf{S_r}. \tag{20}$$

### 3.2. Data and Dictionary Detection

Once we get the compressed and watermarked signal $\mathbf{Y}$ or $\mathbf{y}_i$, it is transmitted to the receiver, thus we get the received signal $\mathbf{Y}'$ or $\mathbf{y}'_i$. The received signals along $\mathbf{y}'_i$ are $\mathbf{U}'_\mathbf{r}$ and $\mathbf{V}'_\mathbf{r}$ as described in Subsection 2. One can choose whether to decompress the signal or to extract the watermark. Anyway, to decompress the signal, we need $\mathbf{A}$ or $\mathbf{p}_{t_i}$ using (22) for reconstructing $\mathbf{y}'_i$ to get $\widehat{\mathbf{S}}_\mathbf{r}$. It is clear that,

**Table 2.** Embedding Process

| | |
|---|---|
| Step 1: | Read a host signal $x(n)$ and transform it into frequency domain by DCT $L$-point obtaining $X(k)$ |
| Step 2: | Reshape $X(k)$ in $L$ sample it to 2-D square matrix producing $\mathbf{X}$ with size $M \times M$ |
| Step 3: | Decompose $\mathbf{X}$ to $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ using SVD |
| Step 4: | Reduce Matrix Size of $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}$ with rank $r$ to $\mathbf{U_r}$, $\mathbf{S_r}$, and $\mathbf{V_r}$ |
| Step 5: | Generate $\mathbf{A}$ matrix containing $p$ Hadamard sequences by mapping each multi watermark bits to an associated random Hadamard sequence using (13) |
| Step 6: | Apply CS acquisition to $\mathbf{A}$ and $\mathbf{S_r}$ by (4) producing $\mathbf{Y}$ |
| Step 7: | Transmit the compressed signal with hidden data represented using $\mathbf{U_r}$, $\mathbf{Y}$ and $\mathbf{V_r}$ |

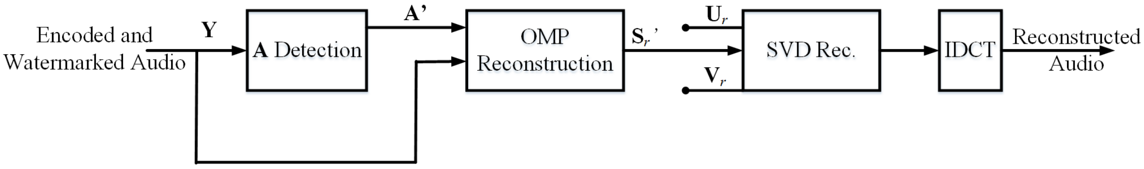


**Figure 2.** Watermark Detection and Audio Decoding

either to extract the watermark or to decompress the signal, extracting $\mathbf{A}$ from $\mathbf{y}_i'$ is the first thing to be applied in the receiver since the compression and the data hiding process is blind. Once we get $\mathbf{A}$, then we can extract the data or we can reconstruct $\mathbf{y}_i'$ with detected $\mathbf{A}$ to obtain $\widehat{\mathbf{S}}_\mathbf{r}'$ using (37), (39), (40) and (41). Thus, we can use SVD reconstruction to $\widehat{\mathbf{S}}_\mathbf{r}'$, $\mathbf{U}_\mathbf{r}'$ and $\mathbf{V}_\mathbf{r}'$ for obtaining a square matrix or $\mathbf{X}_\mathbf{r}'$ using (7). Finally, we get the reconstructed signal $\mathbf{x}'$ by converting the 2 dimensions matrix $\mathbf{X}_\mathbf{r}'$ to the vector $\mathbf{x}'$.

**Table 3.** Detection and Reconstruction Process

| | |
|---|---|
| Step 1: | Detect $t_i$ from $\mathbf{Y}'$ using (22) for extracting the hidden data |
| Step 2: | Associate detected $t_i$ to $\mathbf{p}_{ti}$ and form $\widehat{\mathbf{A}}$ using (13) |
| Step 3: | Reconstruct $\mathbf{Y}'$ using $\widehat{\mathbf{A}}$ by (37), (39), (40) and (41) to obtain $\mathbf{S}_\mathbf{r}'$ |
| Step 4: | Reconstruct $\mathbf{U_r}$, $\mathbf{S}_\mathbf{r}'$, and $\mathbf{V_r}$ by SVD reconstruction to obtain decompressed signal in 2-D matrix $\mathbf{X}_\mathbf{r}'$ by (7) |
| Step 5: | Reshape 2-D matrix $\mathbf{X}_\mathbf{r}'$ to 1-D matrix obtaining $X'(k)$ |
| Step 6: | Transform $X'(k)$ to time domain by IDCT $L$-point obtaining the reconstructed signal $x'(n)$ |

For $\mathbf{p}_{t_i}$ detection, we need to correlate $\mathbf{y}_{t_i}'$ to $\mathbf{p}_j^T$ as

$$K_{ij} = \left|\mathbf{y}_{t_i}'\mathbf{p}_j^T\right|, \tag{21}$$

where $i = 1, 2, .., p$ and $j = 1, 2, .., N_p$. From (21), there is an index of $j$ whose the correlation $K_{ij}$ is the highest, that is $j = t$. Thus, the formula to detect the correct index of Hadamard sequence embedded into $\mathbf{y}_{t_i}'$ is

$$t_i = \underset{j \in \{1,2,...,N_p\}}{\operatorname{argmax}} \left|\mathbf{y}_{t_i}'\mathbf{p}_j^T\right|. \tag{22}$$

Since we can detect $t_i$, we decode the detected Hadamard code to the associated watermark bits according to one to one mapping between the index, the Hadamard code, and the associated watermark bits. For detection proving needs, Assume there is no attack, then $\mathbf{y}'_{t_i}$=$\mathbf{y}_{t_i}$. Thus, (21) is

$$K_{ij} = \left| \mathbf{y}_{t_i} \mathbf{p}_j^T \right| . \tag{23}$$

Substituting (20) into (23) results in

$$K_{ij} = \left| \mathbf{p}_{t_i} \mathbf{S_r} \mathbf{p}_j^T \right| . \tag{24}$$

Assume that $t_i = j$, thus $\mathbf{p}_{t_i} = \mathbf{p}_j$ then (24) is an autocorrelation as

$$K_a = \left| \mathbf{p}_j \mathbf{S_r} \mathbf{p}_j^T \right| . \tag{25}$$

Assume that $\mathbf{p}_j$ consists of such elements as

$$\mathbf{p}_j = \begin{bmatrix} p_{j_1} & p_{j_2} & \cdots & p_{j_r} \end{bmatrix} , \tag{26}$$

therefore (24) becomes

$$K_a = \left| \begin{bmatrix} p_{j_1} \\ p_{j_2} \\ \vdots \\ p_{j_r} \end{bmatrix}^T \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \begin{bmatrix} p_{j_1} \\ p_{j_2} \\ \vdots \\ p_{j_r} \end{bmatrix} \right| . \tag{27}$$

By a matrix multiplication operation, (27) is described as

$$K_a = \left| p_{j_1}^2 \sigma_1 + p_{j_2}^2 \sigma_2 + \cdots + p_{j_r}^2 \sigma_r \right| = \left| \sum_{i=1}^{r} p_{j_i}^2 \sigma_i \right| . \tag{28}$$

Since $\sigma_i > 0$ and $p_{j_i}^2 \gg 0$ for all $j$ and all $i$, then (28) becomes

$$K_a = \sum_{i=1}^{r} p_{j_i}^2 \sigma_i \gg 0. \tag{29}$$

If $\mathbf{p}_{t_i} = \mathbf{p}_k$ and $\mathbf{p}_{t_i} \neq \mathbf{p}_j$, then (24) is a cross-correlation as

$$\begin{aligned} K_c &= \left| \mathbf{p}_k \mathbf{S_r} \mathbf{p}_j^T \right| \\ &= \left| \begin{bmatrix} p_{k_1} \\ p_{k_2} \\ \vdots \\ p_{k_r} \end{bmatrix}^T \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \begin{bmatrix} p_{j_1} \\ p_{j_2} \\ \vdots \\ p_{j_r} \end{bmatrix} \right| \end{aligned} \tag{30}$$

$$\begin{aligned} &= \left| p_{k_1} p_{j_1} \sigma_1 + p_{k_2} p_{j_2} \sigma_2 + \cdots + p_{k_r} p_{j_r} \sigma_r \right| \\ &= \left| \sum_{i=1}^{r} p_{k_i} p_{j_i} \sigma_i \right| . \end{aligned} \tag{31}$$

Since $\mathbf{p}_k$ is mutually orthogonal with $\mathbf{p}_j$, it is confirmed that $K_a$ is comparable to $K_c$ with the following inequality

$$K_a \gg K_c, \tag{32}$$

which means that the autocorrelation of the same Hadamard sequence is still much higher than the cross-correlation of the different Hadamard sequence on the singular value intervention. It confirms that the Hadamard sequence can be detected successfully, thus from (22) $t_i$ is detected for $t_i = \{t_1, t_2, ..., t_p\}$ then we can obtain the associated watermark bits $\hat{\mathbf{w}}_{t_i} = \{\hat{\mathbf{w}}_{t_1}, \hat{\mathbf{w}}_{t_2}, ..., \hat{\mathbf{w}}_{t_p}\}$, and also all Hadamard sequences $\hat{\mathbf{p}}_{t_i} = \{\hat{\mathbf{p}}_{t_1}, \hat{\mathbf{p}}_{t_2}, ...\hat{\mathbf{p}}_{t_p}\}$ which form $\widehat{\mathbf{A}}$ using (13) and (14) as

$$\widehat{\mathbf{A}} = \frac{1}{p}\begin{bmatrix} \hat{\mathbf{p}}_{t_1} \\ \hat{\mathbf{p}}_{t_2} \\ \vdots \\ \hat{\mathbf{p}}_{t_p} \end{bmatrix}, \tag{33}$$

where $p$ is the row number of $\widehat{\mathbf{A}}$. This procedure assures that there is no dictionary needed to detect the hidden data and also to reconstruct the signal. Since the associated watermark bits $\hat{\mathbf{w}}_{t_i}$ are detected. Thus we can calculate Bit Error Rate (BER) as a robustness parameter. The following equation is a BER formula

$$\text{BER} = \frac{\sum_{i=1}^{L_w} |\hat{w}_i - w_i|}{L_w}, \tag{34}$$

where $w_i$ is the original watermark bit, $\hat{w}_i$ is the detected watermark bit, and $L_w$ is the total number of watermark bits.

### 3.3. Security Model

Hadamard matrix is easily generated as described in (11). Anyone can attempt with the Hadamard matrix to reconstruct the dictionary for detecting the hidden data and also to reconstruct the audio. This leads insecure watermark bits hidden in the host audio, accordingly we apply a procedure to secure the Hadamard matrix as also discussed in [15–17]. The Hadamard matrix is multiplied by -1 at the row and the column of the matrix in a random manner. Denote $l_i \in \{1, r\}$ as an integer random permutation value where $i = 1, 2, ..., N_l$, and $N_l$ is the number of generated integer random permutation value. Denote $\mathbf{H}_s$ as a secured Hadamard matrix, $\mathbf{H}_s(j)$ as a vector from $j$-th row of $\mathbf{H}_s$, $\mathbf{H}_s^T(j)$ as a vector from $j$-th column of $\mathbf{H}_s$, then the security model of Hadamard matrix after initial definition $\mathbf{H}_s = \mathbf{H}_r$, is defined as

$$\begin{aligned} \mathbf{H}_s(l_i) &= -\mathbf{H}_r(l_i) \\ \mathbf{H}_s^T(l_i) &= -\mathbf{H}_r^T(l_i) \end{aligned} \tag{35}$$

The above procedure is repeated $N_l$ times from $l_1$ to $l_{N_l}$. Thus, with the secured Hadamard matrix, (12) is replaced by

$$\mathbf{p}_j = \mathbf{H}_s(j). \tag{36}$$

Note that $\mathbf{H}_s$ is not only needed on the embedding process but also in the detection/extraction process. However, it is not needed to pass $\mathbf{H}_s$ to the detector directly. We only pass $l_i$ as the integer random permutation value to the detector as the security key. By the procedure (35), $\mathbf{H}_s$ can be generated in the detector using $l_i$ as the key. According to [15,16], the modified Hadamard matrix combination using (35) has $(r!2^r)^2$ possibilities. For example, if $r = 16$, the number of modified Hadamard matrix is $1.88 \times 10^{36}$ possibilities. If the simulation needs 1 second to run detection and reconstruction process using 1 Hadamard matrix, then it needs $1.88 \times 10^{36}$ seconds or $5.962 \times 10^{28}$ years using all Hadamard matrix possibilities. It confirms that this proposed security model is appropriate and meets the security requirement for the embedding and compression process.

### 3.4. Signal Reconstruction

Once $\widehat{\mathbf{A}}$ is obtained, $\widehat{\mathbf{S}}_{\mathbf{r}}$ reconstruction is simply solved by Orthogonal Matching Pursuit (OMP) [18], [19]. The reconstruction process is carried out on each column of $\mathbf{Y}$ in sequence with $\widehat{\mathbf{A}}$ as a dictionary. Let $\mathbf{y}_m$ as a vector taken from $m$-th column of $\mathbf{Y}$, then for a general case we can find the row position of the strongest atom as

$$q_m = \underset{i \in \{1,2,\ldots,p\}}{\operatorname{argmax}} \widehat{\mathbf{A}}^T \mathbf{y}_m. \tag{37}$$

For a specific case, i.e., a singular matrix solution as the reconstructed one, the position of the highest atoms are indeed known, then (37) can be simplified as

$$q_m = m. \tag{38}$$

Denote $\mathbf{a}_r$ as a vector taken from $r$-th column of $\widehat{\mathbf{A}}$, then we take a column of $\widehat{\mathbf{A}}$ which makes a strongest atom as

$$\nabla = \mathbf{a}_{q_m}. \tag{39}$$

We reconstruct a non-zero element of $\widehat{\mathbf{S}}_\mathbf{r}$ in column $m$ by

$$s_{q_m} = \left(\nabla^T \nabla\right)^{-1} \nabla^T \mathbf{y}_m. \tag{40}$$

This reconstruction procedure including (37), (39), and (40) is repeated $r$ times with the increment of $m$, thus obtaining

$$\widehat{\mathbf{S}}_\mathbf{r} = \begin{bmatrix} s_{q_1} & 0 & \cdots & 0 \\ 0 & s_{q_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{q_r} \end{bmatrix}. \tag{41}$$

Then, the next step is to form the signal by SVD reconstruction, as described in (7). Thus, finally, we can compute the signal quality.

### 3.5. Noisy Environment

Note that the compressed and watermarked audio in this paper is the coded audio. A human cannot directly listen to the coded audio without decoding it first. It means that the signal processing attacks against the coded audio are not the same as the attacks against the real audio signal. The signal processing attacks against the real audio signal were standardized in the Stirmark benchmark [20]. However, the Stirmark benchmark is not appropriate for the robustness evaluation of this proposed method except for the additive noise attack. The additive noise attack is the signal processing attack in which we can generally use to evaluate the watermarking-compression robustness. In the real situation, this additive noise attack in the receiver happens due to the existing thermal condition of the hardware. In this subsection, we describe mathematically how our proposed method is robust to additive noise attack. If the compressed and watermarked signal $\mathbf{y}_i$ is under an additive noise environment, then (23) becomes

$$\begin{aligned} K_{ij} &= \left|(\mathbf{y}_i + \mathbf{n}_i)\, \mathbf{p}_j^T\right| \\ &= \left|(\mathbf{p}_{t_i} \mathbf{S}_\mathbf{r} + \mathbf{n}_i)\, \mathbf{p}_j^T\right|. \\ &= \left|\mathbf{p}_{t_i} \mathbf{S}_\mathbf{r} \mathbf{p}_j^T + \mathbf{n}_i \mathbf{p}_j^T\right| \end{aligned} \tag{42}$$

Assume $\mathbf{p}_{t_i} = \mathbf{p}_j$ then (42) becomes

$$K_{ij} = \left|\mathbf{p}_j \mathbf{S}_\mathbf{r} \mathbf{p}_j^T + \mathbf{n}_i \mathbf{p}_j^T\right|. \tag{43}$$

Because $\mathbf{n}_i$ is independent to $\mathbf{p}_j^T$, thus $\mathbf{p}_j \mathbf{S}_\mathbf{r} \mathbf{p}_j^T \gg \mathbf{n}_i \mathbf{p}_j^T$, then (43) becomes

$$K_{ij} \approx \left|\mathbf{p}_j \mathbf{S}_\mathbf{r} \mathbf{p}_j^T\right| = K_a. \tag{44}$$

Peer-reviewed version available at *Appl. Sci.* **2020**, *10*, 4338; doi:10.3390/app10124338

Thus, we confirm that the data inserted with the proposed method can be detected even in the additive noise environment. The performance evaluation of the proposed method, when attacked by additive noise, depends on the power ratio between host audio and the additive noise represented by Signal to Noise power Ratio (SNR) with the following formula

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum\limits_{i=1}^{r} y_i^2}{\sum\limits_{i=1}^{r} n_i^2}\right) \tag{45}$$

where $i$ is row index at $y$ and $n$, $y_i$ is the signal after compressed using CS at row $i$, $n_i$ is noise at row $i$ and $r$ are the number of rows from $y$.

## 4. Discussion

In this paper, there is more than one work to do in the signal processing environment. The first work is to encode the watermark into the secure Hadamard code. The second work is to make the host audio to be a sparse signal. The third work is to hide the coded watermark into the sparse signal by CS acquisition. Thus, there are two objects for performance analysis, such as the detected watermark and the reconstructed audio from the detected sparse signal. From the embedded watermark relative to the length of the host audio, we can calculate the watermark payload, as described in (17). We can also calculate the CR of sparse technique and CS performance as described in (6) from the host audio length relative to the coded and compressed audio.

Mathematically, we can simply determine the trade-off parameters between the watermark payload and the CR as presented in (17) and (6) respectively. In (17) and (6), there are three same parameters affecting the payload and the CR, such as $M$, $r$ and $p$, where $M$ is the square root of the host audio length or the row/column number of the diagonal matrix ($\mathbf{S}$), $r$ is the row/column number of the truncated diagonal matrix ($\mathbf{S_r}$) and $p$ is the sample number of the compressed signal or the row number of the output of CS acquisition ($\mathbf{Y}$). First, we can see that $p$, $r$ and $M^2$ have different position in (17) and (6). In (17) $p$ and $r$ position are in the numerator which means the decrease of $p$ and $r$ cause the lower payload. In (6) $p$ and $r$ position are in the denumerator which means the decrease of $p$ and $r$ cause the higher CR. Parameter $M^2$ also has different position. This case certainly is a trade-off between payload and CR which we can find the moderate value of $p$ and $r$ to produce high payload and high CR.

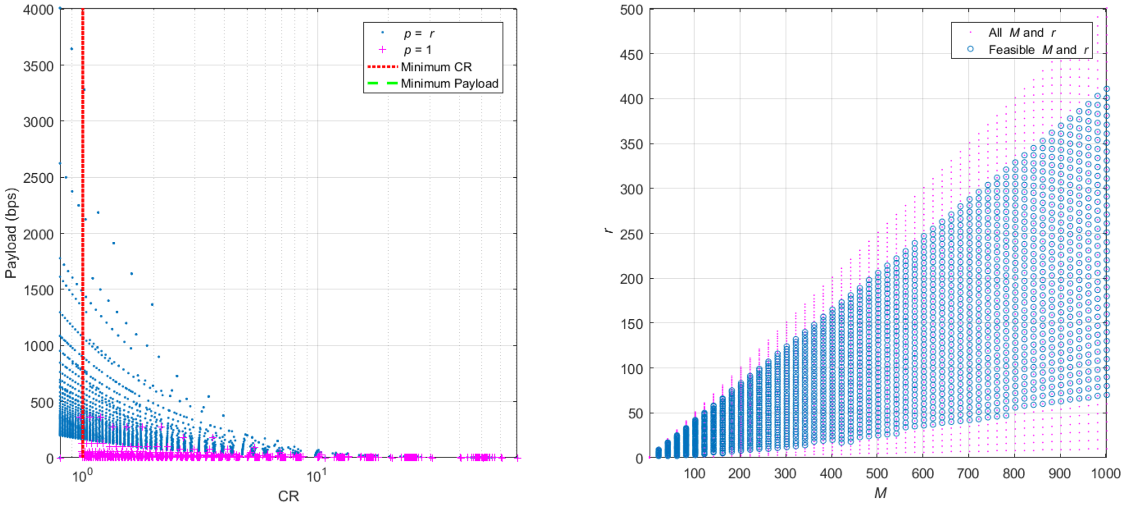
The relation between three parameters $p$, $r$ and $M$ is such that $p \leq r < M$. Referring to (6), the above relation causes the denumerator $pr \ll 2Mr$ if $M$ has a high value, thus

$$\mathrm{CR} \approx \frac{M^2}{2Mr} = \frac{M}{2r}. \tag{46}$$

Note that CR for compression must be more than 1, thus $M/2r > 1$ or $r < M/2$. This means that the minimum truncation for compression is applied at a half of diagonal matrix $\mathbf{S} \in \mathbb{R}^{M\times M}$ obtaining $\mathbf{S_r} \in \mathbb{R}^{\frac{M}{2}\times\frac{M}{2}}$. Consequently, the relation of the three parameters becomes

$$p \leq r < \frac{M}{2}. \tag{47}$$

Thus we can exploit those three parameters in the above relation. Next, we find possible $p$ and $r$ value such that (17) reaches the maximum payload. The position of parameter $p$ and $r$ are in the numerator of (17), thus $r$ should be set to maximum value or $M/2$ in order to obtain maximum payload and $p$ should be set to approximate to $r$. Certainly, setting $r$ to the maximum value or $M/2$ obtains minimum CR, then we have to be careful setting $r$ parameter since it controls the trade-off between $C$ and CR. Due to its position, $p$ parameter should be to the maximum value for reaching the maximum payload. The maximum value of $p$ is $r$. If $p = r$, then CS acquisition, as described in (4), produces an output



**(a)** Payload versus CR with variable $r$ and $M$

**(b)** $r$ versus $M$ for $p = r$ case

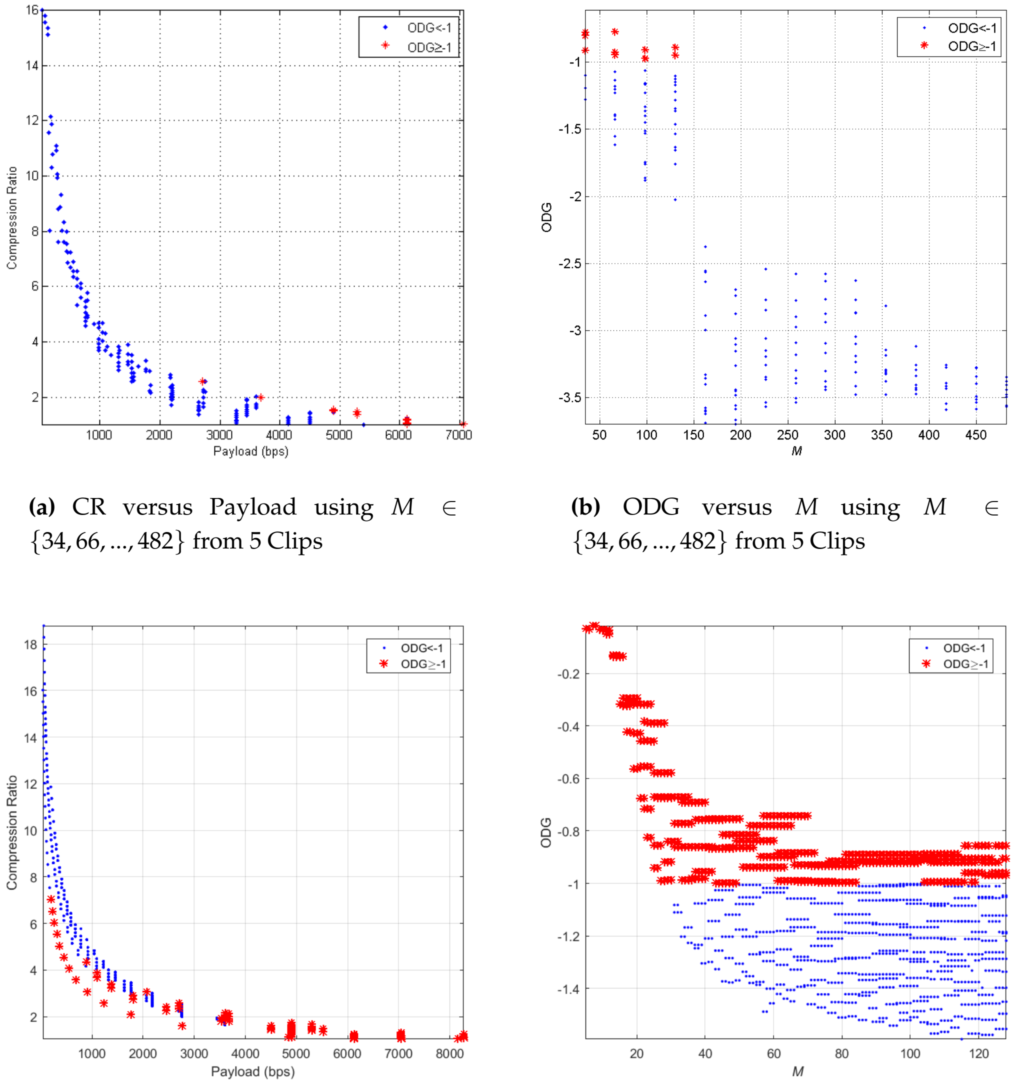
**Figure 3.** Finding feasible $M$ and $r$ to obtain payload>20 bps and CR>1

with the same size to the input of CS. This condition is still acceptable when CR from (6) is more than 1. CS acquisition still contributes to the watermarking process.

Figure 3a displayed the payload versus CR with $M \in \{34, 66, 98, \cdots, 482\}$ and $r \in \{0.01M, 0.02M, \cdots, 0.5M\}$. All possibilities of the $r$ and $M$ combination with the restriction (47) are plotted as the magenta dots in Figure 3b. Blue dots in Figure 3a means the mapping between payload using Equation (17) and CR using Equation (6) where $p = r$, whereas magenta plus signs means the mapping between the payload and CR where $p = 1$. The red vertical dotted line means the minimum CR or 1. The green horizontal dash line means the minimum payload or 20 bps [21]. Thus the area with feasible payload and CR is the right side of the red vertical dotted line and the top side of the green horizontal dash line. We see that many blue dots have a higher payload and CR than the magenta plus signs, which means the payload and CR with $p = r$ has many possibilities to reach much higher ones than the payload and CR with $p = 1$. The payload and CR mapping displayed in the blue dots where payload > 20 bps and CR > 1 in Figure 3a are obtained from $r$ and $M$ in blue circle in Figure 3b, thus we set $p = r$ for the experiment in the next section where $r$ and $M$ combination values are selected from the blue circle in Figure 3b.

## 5. Experimental Result

We assess several evaluations in this section by simulations. The evaluation aspects of the proposed method include audio quality, security, watermark quality, watermark payload, and compression ratio level aspect. The simulations run on ASUS notebooks using Matlab with the following specifications, Advanced Micro Devices (AMD) Fx with 12 compute cores, 16 GB Random Access Memory (RAM), and Windows 10 operating system. There are 50 mono audio host files as the clips tested with the different genres of music, sampling rate 44.1 kHz and 16 bits audio quantization. All clips are in the original wave files and licensed as the free audio files for research [22]. The simulation output in this section shows the average of the simulation result. The evaluated performance parameters are such as the audio quality, the watermark robustness, the watermark payload, and CR. Objective Difference Grade (ODG) represents the audio quality using Perceptual Evaluation of Audio Quality (PEAQ) [23]. Parameter $C$ represents the watermark payload in bps as described at (16). Parameter BER represents the watermark robustness in (34). CR represents the compression ratio, as explained in (6).



**(a)** CR versus Payload using $M \in \{34, 66, ..., 482\}$ from 5 Clips

**(b)** ODG versus $M$ using $M \in \{34, 66, ..., 482\}$ from 5 Clips

**(c)** CR versus Payload using $M \in \{5, 6, ..., 128\}$ from 50 Clips

**(d)** ODG versus $M$ using $M \in \{5, 6, ..., 128\}$ from 50 Clips

**Figure 4.** ODG in relation with $M$, payload and CR

We measure the audio quality between the original host audio and the reconstructed audio. The reconstructed audio quality is affected by two factors, such as the truncation of the diagonal matrix and the CS acquisition. The truncation of the diagonal matrix gives worse quality to the audio than the CS acquisition due to the loss of the audio signal information. The audio quality represented by ODG has a range from -4 to 0, which -4 means the worst audio quality or the distortion is very annoying, -3 means the distortion is annoying, -2 means the distortion is slightly annoying, -1 means the distortion is perceptible but not annoying and 0 means the best audio quality or the distortion is imperceptible [23].

### *5.1. Audio Quality Performance in Relation with $r$, $M$, Payload and Compression Ratio*

From Section 4, we select $M$ and $r$ values to obtain CR $> 1$ and payload $> 20$ bps using $p = r$ as displayed in Figure 3b with the blue circle symbol. Using the selected $M$ and $r$ from $M \in \{34, 66, 98, \cdots, 482\}$ and $r \in \{0.01M, 0.02M, \cdots, 0.5M\}$, we apply the simulation on 5 clips as the hosts. The simulation consists of the embedding process, the data detection process, and the audio reconstruction process. It calculates the BER between the detected watermark and the original watermark, and it finally calculates the audio quality from the reconstructed audio in the ODG performance parameter. The simulation results are displayed in Figure 4a and 4b. From the simulation using all combination parameter $M$ and $r$ with 5 clips, we get a perfect watermark detected without any errors or BER=0 on average. Figure 4a shows the trade-off relation between CR and payload with a negative exponential relation. Red star symbols mean the mapping between CR and payload with ODG $\geq$ -1, while blue dot symbols mean the mapping between CR and payload with ODG < -1. We also plot the blue dots and the red stars in Figure 4b, in the relation between ODG and $M$. We can say that the longer the length of audio processed for embedding and compression causes the worse the reconstructed audio quality. For the above case with 5 selected clips, good reconstructed audio quality or ODG$\geq$ -1 are obtained when $M$<128 samples with certain values of $r$.

**Table 4.** 10 Highest ODG

| $r$ | $M$ | ODG | C | CR |
|---|---|---|---|---|
| 2 | 8 | -0.02 | 2756.25 | 1.60 |
| 2 | 7 | -0.02 | 2756.25 | 1.60 |
| 2 | 5 | -0.03 | 4833.33 | 1.11 |
| 2 | 10 | -0.03 | 1764 | 2.08 |
| 2 | 9 | -0.03 | 1764 | 2.08 |
| 2 | 6 | -0.03 | 4900 | 1.12 |
| 3 | 12 | -0.04 | 5512.50 | 1.45 |
| 3 | 11 | -0.04 | 5512.50 | 1.45 |
| 3 | 10 | -0.04 | 5512.50 | 1.45 |
| 2 | 12 | -0.05 | 1225 | 2.57 |

**Table 5.** 10 Highest Payload

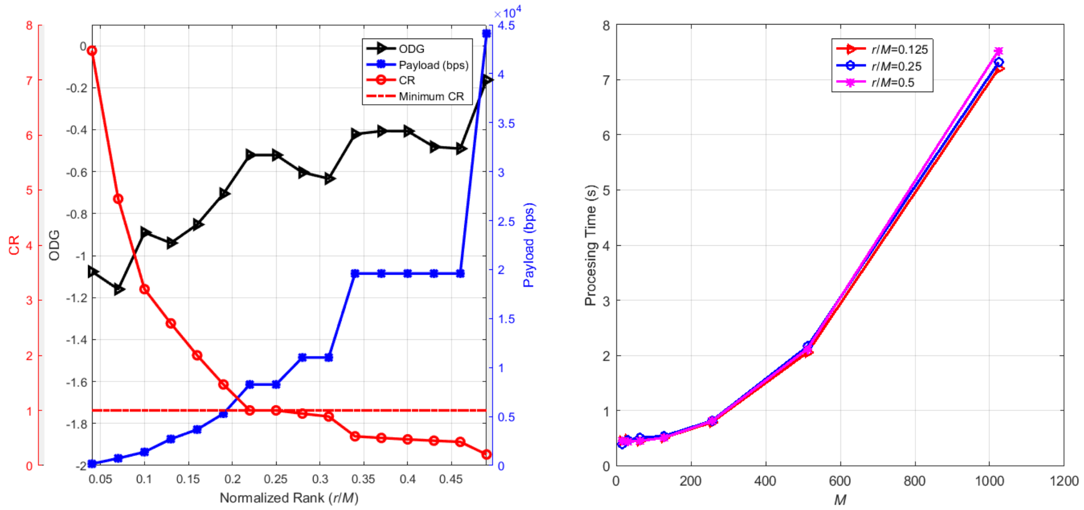
| $r$ | $M$ | ODG | C | CR |
|---|---|---|---|---|
| 5 | 20 | -0.29 | 8268.75 | 1.23 |
| 5 | 19 | -0.29 | 8268.75 | 1.23 |
| 5 | 18 | -0.29 | 8268.75 | 1.23 |
| 5 | 17 | -0.29 | 8268.75 | 1.23 |
| 5 | 16 | -0.29 | 8268.75 | 1.23 |
| 6 | 24 | -0.32 | 8268.75 | 1.14 |
| 6 | 23 | -0.32 | 8268.75 | 1.14 |
| 6 | 22 | -0.32 | 8268.75 | 1.14 |
| 6 | 21 | -0.32 | 8268.75 | 1.14 |
| 6 | 20 | -0.32 | 8268.75 | 1.14 |

**Table 6.** 10 Highest Compression Ratio

| $r$ | $M$ | ODG | C | CR |
|---|---|---|---|---|
| 2 | 30 | -0.99 | 196 | 7.03 |
| 2 | 29 | -0.99 | 196 | 7.03 |
| 2 | 27 | -0.99 | 225 | 6.53 |
| 2 | 28 | -0.99 | 225 | 6.53 |
| 2 | 25 | -0.94 | 260.95 | 6.04 |
| 2 | 26 | -0.94 | 260.95 | 6.04 |
| 2 | 23 | -0.82 | 306.25 | 5.54 |
| 2 | 24 | -0.82 | 306.25 | 5.54 |
| 2 | 22 | -0.68 | 364.47 | 5.04 |
| 2 | 21 | -0.68 | 364.46 | 5.04 |

The required $M$ parameter does not have to be large until 482 samples, but only up to 128 samples to achieve audio quality with ODG $\geq$-1. Figure 4b shows the results. Also, large $M$ values have a long impact on the time processing of the insertion, detection, and reconstruction. Therefore, we apply the same simulation as the simulation displayed in Figure 4a and 4b using more detail $M$ and $r$, i.e., $M \in \{5, 6, ..., 128\}$, $r \in \{1, 2, ..., 64\}$ which is similar to $r \in \{0.0156M, 0.0234M, ..., 0.5M\}$ and 50 clips. We average the audio quality results from 50 clips, and all watermarks are perfectly detected. The simulation results are displayed in Figure 4c and 4d. From Figure 4d, there are much more options of $M$ from 5 to 128 obtaining the results with ODG $\geq$ -1. The simulation as displayed in Figure 4c, also obtains the high CR (up to 7.03), and the high payload (up to 8296 bps). To explore which $M$ and $r$ obtaining the above result, we also capture the simulation results into the table. Table 4, 5 and 6 respectively display 10 highest ODG, payload and CR with certain $M$ and $r$. This simulation results generally show that we can control the audio quality, payload, and CR by adjusting the $M$ and $r$ parameters.

We apply the simulation using 50 clips with $M = 32$ and $r \in \{1, 2, ..., 16\}$ which is similar to $r \in \{0.03M, 0.06M, ..., 0.5M\}$ to see how the audio truncation affects the performance parameters. Figure 5a displays the simulation result. This case also produces perfect detected watermark or BER=0 on average. Three performance parameters, i.e., ODG, CR, and payload as y-axis, are displayed in one figure after averaged, and the x-axis is the normalized rank or $r/M \in \{0.03, 0.06, ..., 0.5\}$. The black line with the right triangle symbol shows the average ODG producing -1.16 to -0.16. The blue line with a square symbol shows the payload of an embedded watermark in bps, obtaining 172.26 to 44100 bps. The red line with a circle symbol means the CR of the encoded audio resulting from 0.20 to 7.53. The red horizontal line with the dashdot symbol means the minimum CR or CR=1. We can see that increasing the normalized rank represented by $r/M$, raises the ODG and the watermark payload but lower the CR of encoded audio. If the CR with the red line and circle symbol is less than the minimum CR, then it means the CS process does not compress the audio signal overall; instead, it increases the length of the encoded signal. In this case, we can select the normalized rank less than 0.2 or $r/M \leq 0.2$



**(a)** Three performance parameters affected by $r/M$

**(b)** Computational Time

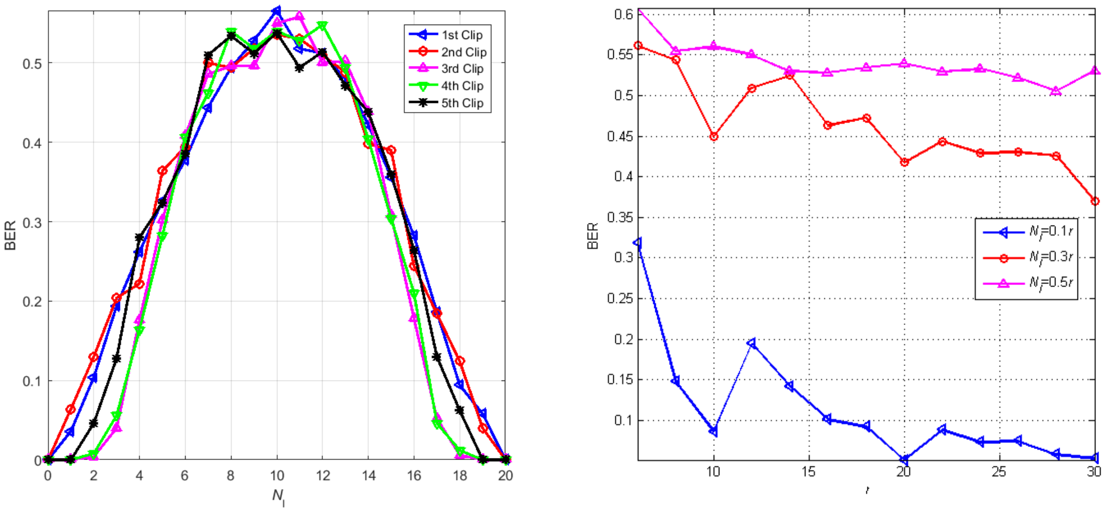
**Figure 5.** Three performance parameters affected by $r/M$ and Computational Time

to keep the CR to be more than 1. In more detail, we can limit minimum $r/M$ such that ODG>-1, i.e., $r/M \geq 0.1$. Thus, the selected range of normalized rank is $[0.1, 0.2]$ obtaining the watermark payload with the range $[729, 5292]$ bps, the compressed ratio with the range $[1.47, 4.84]$ and ODG with the range $[-0.94, -0.74]$. The $r/M$ restriction for this case keeps the reconstructed audio in good audio quality with high payload and CR>1.

### 5.2. Complexity and Computational Time

The major components of the proposed data hiding and compression method in this paper consist of DCT, the multi bits SS mapping, Singular Value Decomposition, and CS acquisition process in the embedding and multi bits SS de-mapping, SVD reconstruction, audio decoding via CS reconstruction and IDCT. Each component has a different complexity. The SVD process to obtain $\mathbf{U} \in \mathbb{R}^{M \times M}$, $\mathbf{S} \in \mathbb{R}^{M \times M}$ and $\mathbf{V} \in \mathbb{R}^{M \times M}$ from $\mathbf{X} \in \mathbb{R}^{M \times M}$ has a complexity of $O(M^3)$ [24]. When we need to get $\mathbf{X}$ from $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$ as (3), its complexity is $O(M^{2.37})$ [25]. DCT and IDCT described in (8) and (9) has a complexity of $O(N_p^2)$ where $N_p$ is the number of the DCT point and $N_p = M$ in this case. The CS acquisition in (4), which is also the multi bits SS embedding, has a complexity of $O(pr^2)$. The multi bits SS detection, as described in 22, has a complexity of $O(r^3)$. Finally, the audio reconstruction by OMP approach in (40) has a complexity of $O(p^2r)$. Due to the relation $p \leq r < M$, the highest computational cost is found in the Singular Value decomposition, i.e., $O(M^3)$, thus the complexity of overall components is dominated by the SVD. This finding confirms the reason to use the lower $M$ value. However, we still need to check the computational time by the simulation to find out a proper $M$ value to avoid a very long processing time.

We apply the simulation to find out the computational time, which should represent the complexity of the embedding and the detection stage. In the simulation, we apply parameter $M$ from 16 to 1024 with multiple of power of 2, parameter $r = 0.125M$, $r = 0.25M$ and $r = 0.5M$. We use 10 clips in the simulation, and we average the time processing result. The result is displayed in Figure 5a. The processing time exponentially increases when $M$ raises. Parameter $r/M$ has no significant impact on the computational time. From this figure, lower $M$ is recommended due to the low computational time. Moreover, as confirmed in Subsection 5.1, the lower $M$ has a significant impact on the reconstructed audio quality.



**(a)** The Effect of $N_l$ to BER

**(b)** The Effect of $r$ to BER

**Figure 6.** BER in relation with $N_l$ and $r$ Using Different Hadamard Matrix Between Encoding and Decoding

### 5.3. Security Analysis

In Section 3.3, there are two parameters having impact to the model security, i.e., $N_l$ as the number of generated integer random permutation value and $r$ as the row and column number of the diagonal matrix after truncated, $\mathbf{S_r}$. The original Hadamard matrix is denoted as $\mathbf{H}_r$, and the secured Hadamard matrix is denoted as $\mathbf{H}_s$. We apply the simulation using vary $r$ and $N_l$ to understand how much $r$ and $N_l$ affect the security performance. In the real situation, one can try to break the security model by using the original Hadamard matrix for detecting the watermark and reconstructing the audio due to the simplicity of the Hadamard matrix generation. With the secured Hadamard matrix in the encoder, we apply the decoding by the original Hadamard matrix to analyze the strength of the security model. If the security model works well, the detection watermark should ideally be damaged, or the BER should be near to 0.5.

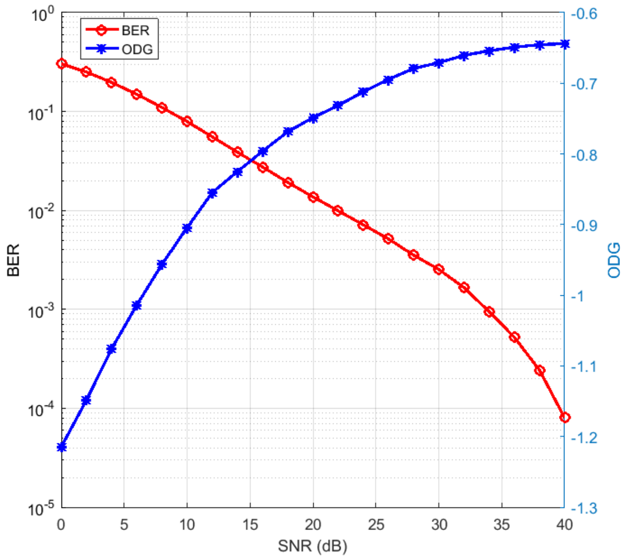
In the simulation, we assume $p = r = 20$ and $M = 128$ samples. $N_l$ varies from 0 to $r$. Parameter $N_l$ is zero, meaning that $\mathbf{H}_s = \mathbf{H}_r$. We use 5 clips for analysis by calculating the average BER after the watermark detection process. We apply the simulation in 100 iterations for each clip. The simulation result is shown in Figure 6a. The worst detected watermark is obtained when $N_l$ is a half of $r$ and the perfect watermark is detected when $N_l = 0$ and $N_l = r = 20$. We can limit accepted minimum BER for restricting the value of $N_l$. We choose BER=0.4 as the safe minimum BER because we can still interpret the digital visualization from the detected watermark with BER<0.3 [26]. Therefore, we choose $N_l > 6$ or generally $N_l > 0.3r$ as the minimum value of $N_l$ and $N_l < 14$ or generally $N_l < 0.7r$ as the maximum value of $N_l$ to keep the detected watermark uninterpretable when one try to detect the watermark by the original Hadamard matrix.

Figure 6b shows the relation between BER and $r$ and comparing the detected watermark quality using the different $N_l/r$. The simulation was applied in 50 clips via 10 iterations for each clip. The range of $r$ is $[6, 30]$. The worst watermark is detected when $N_l/r = 0.5$. The detected watermark quality is better when $N_l/r$ decreases and as the value of $r$ increases. When $N_l/r = 0.3$, most of the BER values are more than 0.4. This result confirms the restriction $N_l$ in the range $[0.3r, 0.7r]$.

### 5.4. Noisy Environment

In the noisy environment, our proposed method is robust to additive noise attack as confirmed mathematically in Section 3.5. Nevertheless, it is necessary to know how robust the method if the additive noise attacks the encoded audio by simulation. We analyze the detected watermark quality represented by BER and the reconstructed audio quality represented by ODG as two performance parameters affected by the additive noise. In the simulation we use 50 clips with 50 iterations for each clip, $M = 23$, $r = 6$ and $p = r$. The additive noise parameter or the input parameter for the simulation is SNR, as described in (45), whose range is 0 to 40 dB. ODG and BER as the performance parameters obtained are averaged before displayed in Figure 7. Decreasing the noise power or increasing the SNR raise the reconstructed audio quality or ODG and the detected watermark quality or BER.

We embed the watermark image with the letters "ITB" and resolution 20 × 35 to understand the interpretation of the value of BER. The detected watermarks are displayed in Table 7 with various BER. We use one selected clip as the audio host using parameter $M = 256, r = 100$ and $p = r$. The original watermark image is shown at the very bottom of Table 7 since its BER is zero. We use the additive noise as the attack with various SNR from 0 to 55 dB. The detected watermark is interpretable as "ITB" when the SNR of the noise is more than 25 dB, or its BER is less than 10%. Thus, the maximum acceptable BER for the detected watermark is up to 10%. In Figure 7, BER less than 10% can be achieved on SNR 10 dB and above. It means that the detected watermark is already interpretable when the noise power is still half of the signal power. Also, ODG is already more than -1. These results confirm the robust proposed method of additive noise. The reconstructed audio is also robust to the additive noise since the ODG already achieves more than -1 when the SNR is still 10 dB.



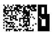
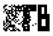
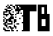
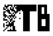
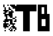
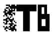
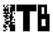
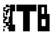
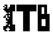
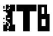
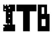
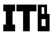
| SNR | BER (%) | Detected Watermark |
|---|---|---|
| 0 | 24.1 |  |
| 5 | 18.7 |  |
| 10 | 17.4 |  |
| 15 | 14.3 |  |
| 20 | 11.7 |  |
| 25 | 8.8 |  |
| 30 | 6.1 |  |
| 35 | 5.7 |  |
| 40 | 5.6 |  |
| 45 | 3.6 |  |
| 50 | 0.4 |  |
| 55 | 0 |  |

**Figure 7 & Table 7.** Additive Noise Effect and The Detected Watermark in Certain SNR

### 5.5. Method Comparison to References

As described in Section 1, there are several references related to this proposed method. We propose a new method with more benefits than the mentioned references. Our proposed method can be used for both audio watermarking or audio steganography with compression due to the controllable parameter between the payload, the audio quality, and the compression ratio. Besides, our proposed method produces the encoding audio, which cannot be attacked by a general signal processing attack, i.e., Stirmark benchmark, except the additive noise as described in Section 3.5. Table 8 displays the comprehensiveness comparison between our proposed method and the previous references, which

**Table 8.** Comprehensiveness Comparison

| Ref. | Hiding Method | Audio Reconstruction | Audio Quality | Robustness | Payload | Compression Ratio |
|---|---|---|---|---|---|---|
| [1] | Watermark Projection | × | × | √ | × | × |
| [2] | Semifragile Zero Watermarking | × | × | √ | × | × |
| [3] | - | √ | √ | × | × | √ |
| [4] | Basis Pursuit Denoising | √ | √ | √ | √ | × |
| Proposed | Multibits Spread Spectrum | √ | √ | √ | √ | √ |

also used CS as the embedding or compression method and the audio as the object to embed or to compress. From the previous reference in Table 8, The reference [3] proposed the audio compression scheme only. The reference [2] proposed the hiding method only. The reference [1] and [4] proposed the hiding method and the compression to the audio but did not analyze all performance parameters.

## 6. Conclusion

In this paper, we propose and report a novel audio watermarking method with CS technique which attempts to insert the watermark into the host audio and simultaneously compresses the audio that has been inserted by the watermark so that the watermarked audio has a smaller size. We also provide the security aspect of this proposed method using a secure Hadamard matrix. The proposed method works well in a noiseless and noisy environment by mathematical derivation. Parameter performance, such as payload, CR, ODG, and BER, are reported in this paper. The experimental result shows that the proposed method presents the high imperceptibility property with payload in the range 729-5292 bps and compression ratio 1.47-4.84. There is a trade-off relation between payload and CR. We can choose the performance, specifically adapting to requirement needs.

**Author Contributions:** conceptualization, Gelar Budiman and Andriyan B. Suksmono; methodology, Gelar Budiman, Andriyan B. Suksmono, Donny Danudirdjo; software, Gelar Budiman; validation, Andriyan B. Suksmono and Donny Danudirdjo; formal analysis, Andriyan B. Suksmono and Donny Danudirdjo; investigation, Gelar Budiman; resources, Gelar Budiman; data curation, Gelar Budiman; writing–original draft preparation, Gelar Budiman; writing–review and editing, Gelar Budiman and Andriyan B. Suksmono; visualization, Gelar Budiman; supervision, Andriyan B. Suksmono and Donny Danudirdjo; project administration, Andriyan B. Suksmono and Donny Danudirdjo; funding acquisition, Andriyan B. Suksmono

**Funding:** This research was funded by the Ministry of Research, Technology and Higher Education of Indonesia in 2020.

**Acknowledgments:** Institut Teknologi Bandung and Telkom University supported this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CS | Compressed Sensing / Compressive Sampling |
| WHT | Walsh Hadamard Transform |
| DCT | Discrete Cosine Transform |
| TLC | Karhunen-Loeve Transform |
| MP3 | Motion Picture Experts Group Audio Layer 3 |
| SVD | Singular Value Decomposition |
| CR | Compression Ratio |
| IDCT | Inverse DCT |
| SS | Spread Spectrum |
| BER | Bit Error Rate |
| OMP | Orthogonal Matching Pursuit |
| SNR | Signal to Noise power Ratio |
| ODG | Objective Difference Grade |
| PEAQ | Perceptual Evaluation of Audio Quality |
| AMD | Advanced Micro Devices |
| RAM | Random Access Memory |

## References

1. Hua, G.; Xiang, Y.; Bi, G. When Compressive Sensing Meets Data Hiding. *IEEE Signal Processing Letters* **April 2016**, *23*, 473–477.
2. Zhaofeng, M.; Xinxin, N.; Yixian, Y. Compressive Sensing-Based Audio Semi-fragile Zero-Watermarking Algorithm. *Chinese Journal of Electronics* **July 2015**, *24*, 492—497. doi:10.1049/cje.2015.07.009.
3. Griffin, A.; Hirvonen, T.; Tzagkarakis, C.; Mouchtaris, A. Single-Channel and Multi-Channel Sinusoidal Audio Coding Using Compressed Sensing. *IEEE Transactions on Audio, Speech, and Language Processing* **July 2011**, *19*, 1382–1395. doi:10.1109/TASL.2010.2090656.
4. Fakhr, M.W. Robust Watermarking Using Compressed Sensing Framework With Application To MP3 Audio. *The International Journal of Multimedia and Its Applications (IJMA)* **December 2012**, *4*, 27–43.
5. Pawlak, M. M-Ary Phase Modulation for Digital Watermarking. *International Journal of Applied Mathematics and Computer Science* **2008**, *18*, 93—-104. doi:10.2478/v10006-008-0009-8.
6. Xiang, Y.; Natgunanathan, I.; Rong, Y.; Guo, S. Spread Spectrum Based High Embedding Capacity Watermarking Method for Audio Signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **December 2015**, *9290*, 2228–2237. doi:10.1109/TASLP.2015.2476755.
7. Xiang, Y.; Member, S.; Natgunanathan, I.; Peng, D.; Hua, G.; Liu, B. Spread Spectrum Audio Watermarking Using Multiple Orthogonal PN Sequences and Variable Embedding Strengths and Polarities. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **March 2018**, *26*, 529–539. doi:10.1109/TASLP.2017.2782487.
8. Sabrina, M.B.; Khaled, R.; Salim, A.; Djamel, C. Spreading codes performances comparison in terms of Bit-Error-Rate in additive white Gaussian noise channel. 2015 4th International Conference on Electrical Engineering (ICEE), 2015, pp. 1–6. doi:10.1109/INTEE.2015.7416718.
9. Yadav, S.K.; Sinha, R.; Bora, P.K.; Formulation, A.P.; Art, B.P. An Efficient SVD Shrinkage for Rank Estimation. *IEEE Signal Processing Letters* **December 2015**, *22*, 2406–2410.
10. Rusu, C.; Dumitrescu, B. Stagewise K-SVD to Design Efficient Dictionaries for Sparse Representations. *IEEE Signal Processing Letters* **2013**, *19*, 631–634.
11. Zhang, S.; Zhu, Y.; Dong, G.; Kuang, G. Truncated SVD-Based Compressive Sensing for Imaging With Uniform / Nonuniform Linear Array. *IEEE Geoscience and Remote Sensing Letters* **2015**, *12*, 1853–1857.
12. Wang, X.Y.; Zhao, H. A novel synchronization invariant audio watermarking scheme based on DWT and DCT. *IEEE Transactions on Signal Processing* **2006**, *54*, 4835–4840. doi:10.1109/TSP.2006.881258.
13. Guo, J.; Xie, R.; Jin, G. An Efficient Method for NMR Data Compression Based on Fast Singular Value Decomposition. *IEEE Geoscience and Remote Sensing Letters* **Februari 2019**, *16*, 301–305. doi:10.1109/LGRS.2018.2872111.
14. Strang, G. *Linear Algebra and Its Applications*; Thomson Learning, 2006; pp. 1–497.

15. Ville, D.V.D.; Philips, W.; Walle, R.V.D.; Lemahieu, I.; Member, S.; Van De Ville, D.; Philips, W.; Van de Walle, R.; Lemahieu, I. Image scrambling without bandwidth expansion. *IEEE Transactions on Circuits and Systems for Video Technology* **2004**, *14*, 892–897. doi:10.1109/TCSVT.2004.828325.
16. Pal, S.K. Fast, Reliable Secure Digital Communication Using Hadamard Matrices. 2007 International Conference on Computing: Theory and Applications (ICCTA'07), 2007, pp. 526–532. doi:10.1109/ICCTA.2007.61.
17. Bouguezel, S.; Ahmad, M.O.; Swamy, M.N.S. Binary Discrete Cosine and Hartley Transforms. *IEEE Transactions on Circuits and Systems I: Regular Papers* **2013**, *60*, 989–1002. doi:10.1109/TCSI.2012.2224751.
18. Schnass, K. Average performance of Orthogonal Matching Pursuit ( OMP ) for sparse approximation. *IEEE Signal Processing Letters* **2018**, *25*, 1865–1869. doi:10.1109/LSP.2018.2878061.
19. Chen, Y.; Li, G.; Gu, Y. Active Orthogonal Matching Pursuit for Sparse Subspace Clustering. *IEEE Signal Processing Letters* **2018**, *25*, 164–168.
20. Lang, A.; Dittmann, J.; Spring, R.; Vielhauer, C. Audio Watermark Attacks : From Single to Profile Attacks. MM&Sec '05 Proceedings of the 7th workshop on Multimedia and security. ACM, 2005, pp. 39–50. doi:10.1145/1073170.1073179.
21. Katzenbeisser, S.; Petitcolas, F.A.P. *Information Hiding Techniques for Steganography and Digital Watermarking*; Artech House, Inc: Boston, London, 2000; p. 237.
22. Turetsky, R.; Ellis, D. Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses. *4th International Conference on Music Information Retrieval* **2003**.
23. ITU-R, R. RECOMMENDATION ITU-R BS.1387-1 Method for objective measurements of perceived audio quality. Technical report, International Telecommunications Union, 1998.
24. Cline, A.K.; Dhillon, I.S. *Computation of the Singular Value Decomposition*; CRC Press, 2006.
25. Gall, F.L. Powers of Tensors and Fast Matrix Multiplication. *CoRR* **2014**, *abs/1401.7714*, [1401.7714].
26. Lei, B.; Soon, I.Y.; Tan, E.l. Robust SVD-Based Audio Watermarking Scheme With Differential Evolution Optimization. *IEEE Transactions on Audio, Speech, and Language Processing* **2013**, *21*, 2368–2378.