Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2035747, 15 pages
https://doi.org/10.1155/2020/2035747

Research Article

Robust and Blind Audio Watermarking Algorithm in Dual Domain for Overcoming Synchronization Attacks

Qiuling Wu,[1,2] **Aiyan Qu**,[1] **and Dandan Huang**[1]

[1]*School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China*
[2]*College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Qiuling Wu; wuqiuling@jit.edu.cn

Received 1 June 2020; Revised 9 November 2020; Accepted 11 November 2020; Published 21 November 2020

Academic Editor: Mohamed Shaat

How to effectively resist synchronization attacks is the most challenging topic in the research of robust watermarking algorithms. A robust and blind audio watermarking algorithm for overcoming synchronization attacks is proposed in dual domain by considering time domain and transform domain. Based on analysing the characteristics of synchronization attacks, an implicit synchronization mechanism (ISM) is developed in the time domain, which can effectively track the appropriate region for embedding and extracting watermarks. The data in this region will be subjected to discrete cosine transform (DCT) and singular value decomposition (SVD) in turn to obtain the eigenvalue that can be utilized to carry watermarks. In order to extract the watermark blindly, the eigenvalue will be quantized. Genetic algorithm (GA) is utilized to optimize the quantization step to balance both transparency and robustness. The experimental results confirm that the proposed algorithm not only withstands various conventional signal processing operations but also resists malicious synchronization attacks, such as time scale modification (TSM), pitch-shifting modification (PSM), jittering, and random cropping. Especially, it can overcome TSM with strength from −30% to +30%, which is much higher than the standard of the International Federation of the Phonographic Industry (IFPI) and far superior to the other algorithms in related papers.

1. Introduction

1.1. Related Works. With the rapid development of network and computer technology, people edit, modify, store, and disseminate audio media easily by using various audio editing software [1–3]. While the editing software brings us convenience, it also makes unauthorized users perform a variety of infringements on the audio media, such as malicious tampering, forgery, deletion, and unauthorized distribution. Sometimes, these infringements not only jeopardize the safety of personal property and the credibility of the audio media but also may even endanger national public safety in acute cases [4–6]. How to effectively protect the security of those audio media has become a research hotspot in information security, communication, and some related fields. Robust audio watermarking algorithm pays much attention to improve its ability for preventing watermarks hidden in the audio from being destroyed under complex environments [7, 8], so it must not only be able to withstand the conventional signal processing operations encountered when using those audio media normally but also need to be extremely resistant to many malicious synchronous attacks that may cause the structure of the audio media to change.

Synchronization attacks may cause serious damage to the structure of the audio, resulting in the extraction failure due to the inaccuracy of the embedding region [9–12], so they have become the most challenging attacks in the research of audio watermarking algorithms [13–15]. Hu et al. [16] proposed an audio watermarking algorithm based on lifting wavelet transform. The authors claimed that the algorithm had good robustness to some conventional signal processing attacks and synchronization attacks, and its payload capacity reached 43.07 bps when SNR was over

21 dB. However, it can be seen from the experimental results that the algorithm robustness still needs to be improved when resisting TSM. Xiang and Huang [17] designed an audio watermarking algorithm with a constant watermark synchronization mechanism according to the insensitivity of the histogram shape of audio media. Hu and Chang [18] proposed a self-synchronous audio watermarking algorithm based on discrete wavelet transformation (DWT) and DCT. This algorithm concealed the synchronous signal in the first approximation sub-band and recalibrated the embedding position by extracting the zero-crossing point of the synchronous signal. The experimental results showed that this algorithm was effective for some synchronization attacks but poor for some signal processing attacks. Wang et al. [19] proposed a robust audio watermarking algorithm which utilized the invariance of exponential moment to enhance its robustness. However, it was poor for amplitude scaling and MP3 compression as shown in experimental results. Yuan et al. [20] put forward an audio watermarking algorithm that detected the mel-cepstrum coefficient as a synchronous signal when extracting the watermark in the DWT domain. Wang et al. [21] proposed a robust audio watermarking algorithm based on empirical mode decomposition. In this algorithm, the audio was evenly segmented into numerous fragments, and then each audio fragment was separated into two parts. One part was utilized to embed the synchronization code, and the other part was used to embed the watermark in the residue of higher-order statistics after empirical mode decomposition. If synchronization codes could not be accurately acquired, watermark extraction would fail, which was a fatal shortcoming of this algorithm. Chen et al. [22] proposed an audio watermarking algorithm that embedded the watermark into the low-frequency coefficients of the audio in the DWT domain. This algorithm enhanced its robustness by increasing the embedding depth, but this behaviour also led to the low transparency. In general, audio watermarking algorithms with the ability to resist synchronization attacks must have an effective synchronization mechanism, which can be used to track the embedding position [23, 24]. However, most existing algorithms are usually robust to only one or two of these attacks, and some algorithms even lose robustness to conventional signal processing operations due to their excessive pursuit of robustness to some synchronization attacks. In addition, how to balance the overall performance of the algorithm by optimizing the parameters of the designed algorithm is also an issue with research significance.

1.2. Contributions. Based on the above introduction, we can see that there are still many problems to be solved in antisynchronization attacks. Our contributions in this paper are as follows.

(1) An ISM is developed to effectively search for the appropriate embedding region when embedding watermarks and to automatically track the region where the watermark is located when extracting watermarks. Based on analysing the characteristics of synchronization attacks, it is found that the shape of the voiced frame almost has not changed after being subjected to TSM, so the proposed ISM takes the sample point with the largest amplitude in the voiced frame as the synchronization mark to identify the embedding region and extracting region. When embedding watermarks, the appropriate region will be searched out from the voiced frames by using ISM, and then the data in the chosen embedding region will be further operated to carry watermarks. When extracting watermarks, ISM can automatically track the region where the watermark is located.

(2) GA is utilized to optimize the key algorithm parameter to balance both transparency and robustness. The data in the embedding region will be processed by DCT and SVD in turn to obtain the eigenvalue that can be used to carry watermarks. In order to extract the watermark blindly, the eigenvalue is quantized when embedding or extracting the watermark, so the quantization step is an important parameter, which directly affects the transparency and robustness of the algorithm. We propose an optimal audio watermarking algorithm using GA to further enhance the overall performance of this algorithm.

Besides, this algorithm adopts several additional measures to improve the robustness, such as twice even segmentation to the audio, and the operation that embeds the same watermark into three voiced frames.

The remainder of this paper is organized as follows. In Section 1, we review some related works about the existing audio watermarking algorithms which can overcome synchronization attacks and then introduce our contributions in this proposed algorithm. Section 2 describes the proposed ISM and shows the implementation flow chart in detail. The principle of the proposed audio watermarking algorithm will be elaborated in Section 3, and this section will be divided into four subjects, including the embedding principle, the extracting principle, optimization of the quantization step, and the measure to further improve robustness. Section 4 evaluates the performance of this proposed algorithm and compares their performance with other algorithms in recent years. Finally, Section 5 draws up the conclusion and gives the possible future research task.

2. ISM for Tracking Embedding Region

Synchronization attacks may cause the position of the data in the audio to shift, which may lead to extraction failure because the location of the watermark cannot be obtained accurately [25]. Therefore, it is very important to design an effective synchronization mechanism for tracking the embedding region. If the data in the voiced frames are modified too much, the audio may not be used normally because of the obvious degradation of audio quality, so synchronization attack usually only modifies the data in redundant frames, but not in voiced frames. TSM attacks by 10% and −10% are applied to an audio clip, respectively, and the waveform comparison is illustrated in Figure 1. It

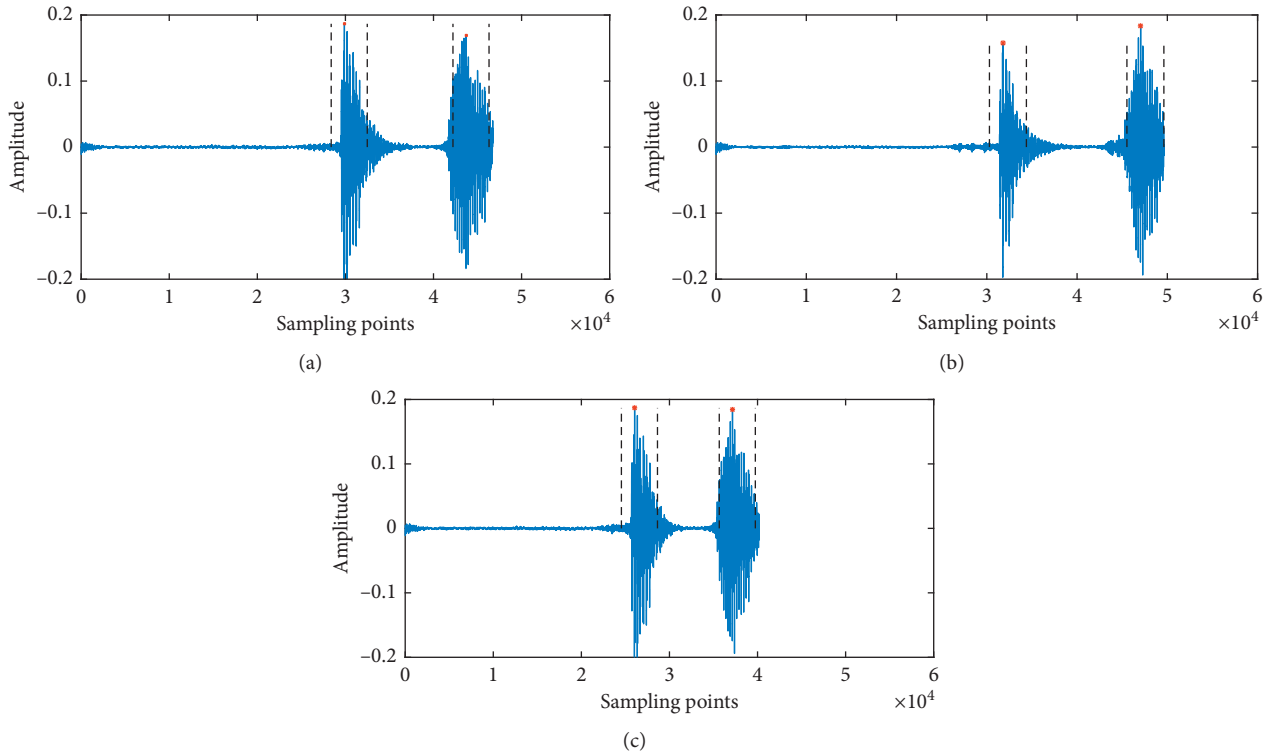

Figure 1: Audio waveform with the searched embedding region under TSM. (a) The embedding regions without attack. (b) The embedding regions under −10% TSM. (c) The embedding regions under +10% TSM.

can be seen from the pictures that the absolute positions of the two voiced frames in this audio clip all have shifted on the time axis, but their shapes do not change much in the process of being stretched in Figure 1(b) or compressed in Figure 1(c), so it is relatively safe to conceal watermarks into these voiced frames.

If the watermark is only embedded in voiced frames and the embedding region in the audio is independent of the absolute position of the audio data on the time axis, it will greatly improve the algorithm's ability to withstand synchronization attacks. As long as the embedding region can be effectively tracked, the watermark will be accurately extracted. Based on the above analysis, an ISM is developed in our study, which can search out the appropriate embedding region when embedding watermarks and can effectively track the extracting region where the watermark is located when extracting watermarks. As shown in Figure 1, the regions between the two red dashed vertical lines are the embedding regions in the two voiced frames, and "*" indicates the synchronization mark which is the position of the tracked sample point with the largest amplitude. It can be observed in Figure 1 that the proposed ISM can more accurately track the appropriate embedding region under TSM.

Figure 2 shows the implementation flowchart of the ISM. Assuming that the length of the voiced frame is N, the length of the region for embedding watermarks is N_e and $N \geq N_e$. The specific implementation process can be described as follows.

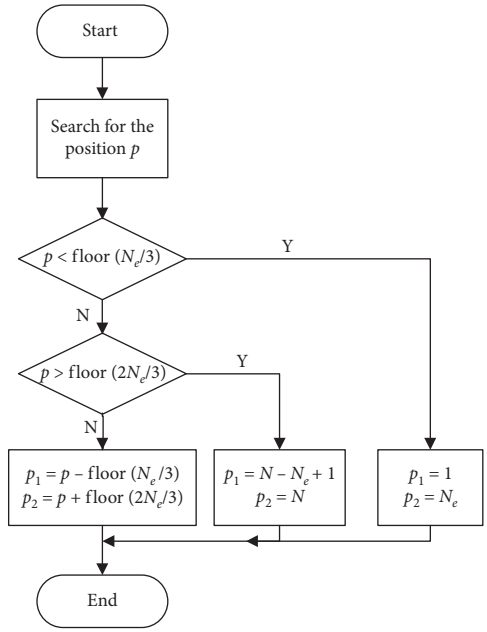

Figure 2: Flowchart of the proposed ISM.

(i) Step 1: extract all the voiced frames with the length of N from the audio.

(ii) Step 2: search for the sample point with the largest amplitude in each voiced frame and record its position as p.

(iii) Step 3: N_e sample points in the surrounding region of p, which are in the range of $[p_1, p_2]$, can be used to carry watermarks, where $p_1 = p - \text{floor}(N_e/3)$ is the starting position in this embedding region, $p_2 = p + \text{floor}(2N_e/3)$ is the ending position, and floor() indicates that the data in brackets should be rounded down.

The embedding region $[p_1, p_2]$ has the following three conditions.

(1) If $p < \text{floor}(N_e/3)$, it indicates that the position p of the sample point with the largest amplitude is closer to the head of this voiced frame, then the starting position of the embedding region should be set as $p_1 = 1$, and the ending position is $p_2 = N_e$.

(2) If $p > \text{floor}(2N_e/3)$, this condition shows that the position p is closer to the end of this voiced frame, then the starting position should be set as $p_1 = N - N_e + 1$, and the ending position is $p_2 = N$.

(3) If $\text{floor}(N_e/3) \le p \le \text{floor}(2N_e/3)$, it indicates that the position p is in the middle part of this voiced frame, then the starting position of the region should be set as $p_1 = p - \text{floor}(N_e/3)$, and the ending position is $p_2 = p + \text{floor}(2N_e/3)$.

3. Principle of the Watermarking Algorithm

3.1. Principle of Embedding Watermarks. The embedding algorithm mainly includes the following several parts. Firstly, the proposed ISM is used to search for the best embedding region. Then, DCT is performed on the data in the embedding region to determine the frequency range for carrying the watermark. Finally, the DCT coefficients in the frequency range are processed by SVD to conceal the watermark by the quantization method. Figure 3 shows the principle diagram of the embedding algorithm.

Suppose that the binary watermark can be expressed in the following formula:

$$W_0\{w_0(i), \quad 1 \le i \le L_w\}, \tag{1}$$

where $w_0(i) \in \{0, 1\}$ and L_w is the length of W_0. In order to improve the security, W_0 should be encrypted before it is concealed into the audio.

Apply logistic mapping formula to generate a chaotic sequence $c(i)$ with the same size as W_0, as shown in the formulas.

$$x_{i+1} = a_0 x_i (1 - x_i),$$
$$c(i) = \begin{cases} 1, & x_i \ge \delta, \\ 0, & x_i < \delta, \end{cases} \tag{2}$$

where $0 < x_i < 1$, $x_1 \in (0, 1)$ is the initial value when $i = 1$, and $\delta \subset [0, 1]$ is a threshold to obtain $c(i)$. The logistic system will be in chaos when $3.5699456 \le \alpha_0 \le 4$.

Exclusive OR operation is performed on W_0 and $c(i)$ to obtain the encrypted information W_1, as shown in formula (3), where $\oplus$ stands for the exclusive OR operator. Triple key $\text{Ch}(x_1, \alpha_0, \delta)$ is the unique correct key to decrypt W_1.

$$W_1 = \{w_1(i) = w_0(i) \oplus c(i), \quad 1 \le i \le L_w\}. \tag{3}$$

Convert W_1 into a two-dimensional array W_2 shown in formula (4), where $w_2(u, v) \in \{0, 1\}$ and $L_w = L_1 \times L_2$. Suppose that A is the original audio with K sample points, as expressed in the formula.

$$W_2 = \{w_2(u, v), \quad 1 \le u \le L_1, \ 1 \le v \le L_2\}, \tag{4}$$

$$A = \{a(j), \quad 1 \le j \le K\}, \tag{5}$$

where $a(j)$ is the amplitude of the j^{th} sample point. Divide A into L_1 audio fragments, namely, A_l $(1 \le l \le L_1)$, and each audio fragment has K_1 sample points, $K_1 = \text{floor}(K/L_1)$.

Then, A can be divided into two parts, namely, Ar and As, where Ar will be used for carrying the watermark, and As does not participate in the embedding process. Ar can be expressed in formula (6), and its size is $L_1 \times K_1$.

$$Ar = \begin{bmatrix} A_1 \\ \vdots \\ A_l \\ \vdots \\ A_{L_1} \end{bmatrix}_{L_1 \times K_1} = \begin{bmatrix} a(1) & a(2) & \cdots & a(K_1) \\ \vdots & & & \vdots \\ a((l-1)K_1 + 1) & a((l-1)K_1 + 2) & \cdots & a(lK_1) \\ \vdots & & & \vdots \\ a((L_1 - 1)K_1 + 1) & a((L_1 - 1)K_1 + 2) & \cdots & a(L_1 K)_1 \end{bmatrix}_{L_1 \times K_1}. \tag{6}$$

The watermark W_2 will be embedded into Ar, that is to say, each audio fragment A_l needs to carry L_2 bits watermark. To prevent the audio quality from decreasing too much, A_l will be divided into several audio frames with the length of N, and only the voiced frame $A_{\max}$ with the largest energy is used to carry the watermark. The proposed ISM is used to track the appropriate embedding region $[p_1, p_2]$ in $A_{\max}$.

We will take the embedding process that embed L_2 bits binary watermark into $A_{\max}$ as an example to illustrate the core embedding scheme. Figures 4 and 5 show the main data and the flowchart of this core embedding scheme.

In our study, DCT is used to determine the frequency range where the watermark is located. Apply DCT on the data between $[p_1, p_2]$ to obtain the DCT coefficient A_{dct} and then get the intermediate frequency coefficient A_{if} from A_{dct} in the range $[b_0 + 1, b_0 + L_{if}]$, where b_0 is the starting position and L_{if} is the length of A_{if}. The frequency range $[f_1, f_2]$ for embedding the watermark can be calculated according to the formulas.

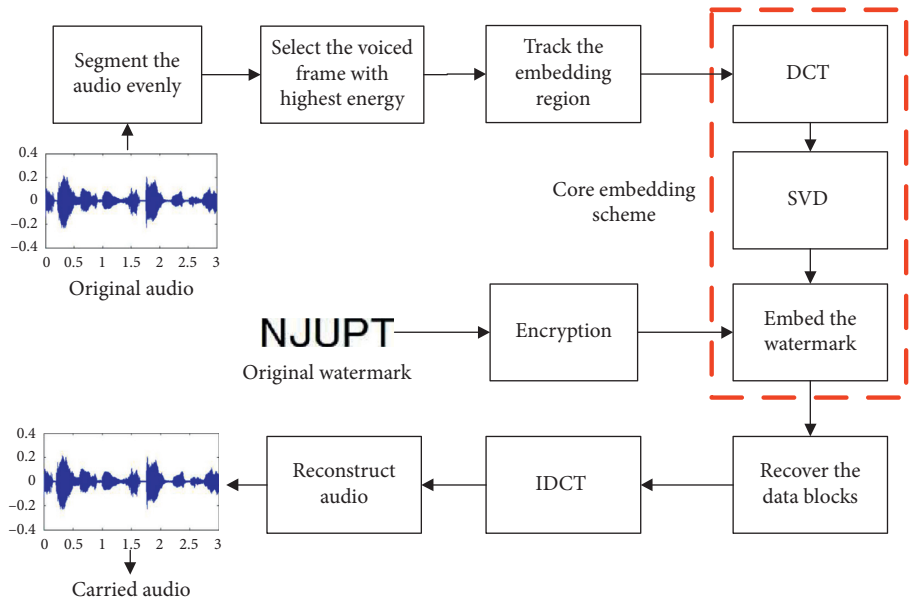

Figure 3: The principle diagram of the embedding algorithm.

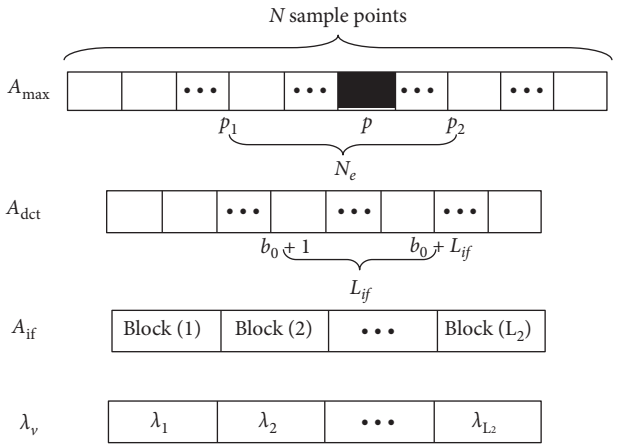

Figure 4: Data in the core embedding algorithm.

$$f_1 = \frac{(b_0 + 1) \times f_h}{N_e} \text{kHz},$$
$$f_2 = \frac{(b_0 + L_{if}) \times f_h}{N_e} \text{kHz}, \tag{7}$$

where f_h is the maximum cutoff frequency of the audio, and its value is usually half of the sampling rate. Divide A_{if} into L_2 data blocks, namely, Block(v) $(1 \le v \le L_2)$, and the length of the data blocks can be calculated as $L_{svd} = \text{floor}(L_{if}/L_2)$. Apply SVD on Block$(v)$ to obtain the eigenvalue λ_v, as shown in formula (8), $U_v = [u_v]$ where is a single element matrix, V_v is an orthogonal matrix with the dimension of $L_{svd} \times L_{svd}$, and S_v is a row matrix with the dimension of $1 \times L_{svd}$ in which only the first element $\lambda_v \neq 0$ and all other elements are equal to 0.

$$\text{Block}(v) = USV^T = [u_v][\lambda_v \ 0 \ \dots \ 0] \begin{bmatrix} v_{11} & \dots & v_{1L_{svd}} \\ \vdots & \ddots & \vdots \\ v_{L_{svd}1} & \cdots & v_{L_{svd}L_{svd}} \end{bmatrix}. \tag{8}$$

According to the stability characteristics of SVD, the eigenvalue λ_v usually does not change greatly when Block(v) changes slightly, so one bit binary watermark can be hidden into one eigenvalue. In order to realize blind extraction, $cc = \text{floor}(\lambda_v/a)$ will be obtained by quantization, where a is the quantization step. If the binary watermark is "0," cc will be modified to be an even number; otherwise, cc will be set as an odd number. The embedding rule is described in Table 1.

Then, the modified eigenvalue λ_v' is shown in the following formula:

$$\lambda_v' = a \times cc \pm \frac{a}{2}. \tag{9}$$

The modified data block Block$'(v)$ can be reconstructed according to the following formula:

$$\text{Block}'(v) = US'V^T. \tag{10}$$

Repeat the above process to modify all eigenvalues, and L_2 bits binary watermark can be concealed into all Block(v) $(1 \le v \le L_2)$. According to the process described above, each row of the binary data in W_2 can be concealed into each A_l $(1 \le l \le L_1)$; finally, W_2 will be completely concealed into A_r. The embedding process can be described as follows.

(i) Step 1: convert W_1 into W_2 with the size of $L_1 \times L_2$.

(ii) Step 2: divide the original audio A into L_1 audio fragments A_l with the same length.

(iii) Step 3: divide A_l into audio frames with the length of N and find out the voiced frame A_{max}.

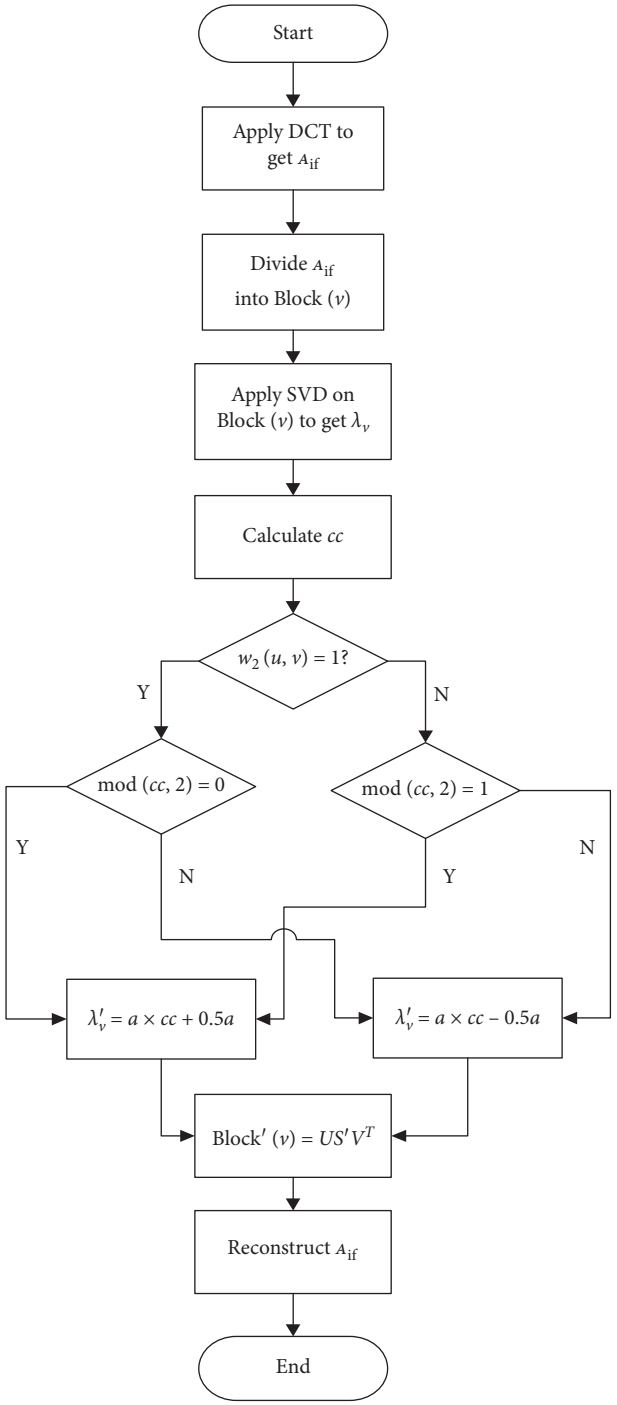

Figure 5: Flowchart of the core embedding algorithm.

(iv) Step 4: search for the embedding region $[p_1, p_2]$ by using ISM in $A_{\max}$.

(v) Step 5: apply DCT on the data in the embedding region to obtain A_{if}.

(vi) Step 6: divide A_{if} into L_2 data blocks Block(v) and perform SVD on Block(v) to obtain $\overset{v}{\lambda}$.

(vii) Step 7: embed L_2 bits binary watermark into $A_{\max}$ according to the embedding rule in Table 1.

(viii) Step 8: repeat Step 4 to Step 7 until all watermark bits are concealed.

(ix) Step 9: recombine all audio fragments to recover the carried audio A'.

3.2. Principle of Extracting Watermarks. The extracting algorithm is the inverse process of the embedding algorithm, and its principle is shown in Figure 6, in which the "core extracting scheme" is the most important part of the whole extracting algorithm. The process of extracting the watermark can be described as follows.

(i) Step 1: divide the carried audio A' into L_1 audio fragments A'_l with the same length.

(ii) Step 2: divide A'_l into audio frames with the length of N and find out the voiced frame $A'_{\max}$.

(iii) Step 3: track the region containing the watermark by using ISM in $A'_{\max}$.

(iv) Step 4: apply DCT on the data in the tracked region to obtain A'_{if}.

(v) Step 5: divide A'_{if} into L_2 data blocks Block$'(v)$ and perform SVD on Block$'(v)$ to obtain λ'_v.

(vi) Step 6: quantify all eigenvalues and judge their parity to obtain L_2 bits binary watermark. The extracting rule can be expressed in formula (11), where $cc' = \text{floor}(\lambda'_v/a)$, $1 \le u \le L_1$, $1 \le v \le L_2$.

(vii) Step 7: repeat Step 2 to Step 6 until all watermark bits are extracted.

(viii) Step 8: decrypt and recover the watermark w'_1 from w'_2.

$$w'_2(u, v) = \begin{cases} 0, & cc' \text{ is even,} \\ 1, & cc' \text{ is odd.} \end{cases} \tag{11}$$

Figure 7 shows the flowchart of the core extracting scheme. In particular, the key parameters in the extracting algorithm should be consistent with the corresponding parameters in the embedding algorithm, including N, N_e, L_1, L_2, b_0, L_{if}, and a.

3.3. Optimization of the Quantization Step. From the watermark embedding principle mentioned above, quantization step is an important parameter, which directly affects the transparency and robustness of the algorithm. In order to balance the algorithm performance, GA is used to search the optimal quantization step intelligently. The fitness function Fitness is constructed with SNR and BER as shown in the formula.

$$\begin{cases} \text{SNR} > \text{SNR}_0, \\ \text{Fitness} = \dfrac{1}{\text{BER}}, \end{cases} \tag{12}$$

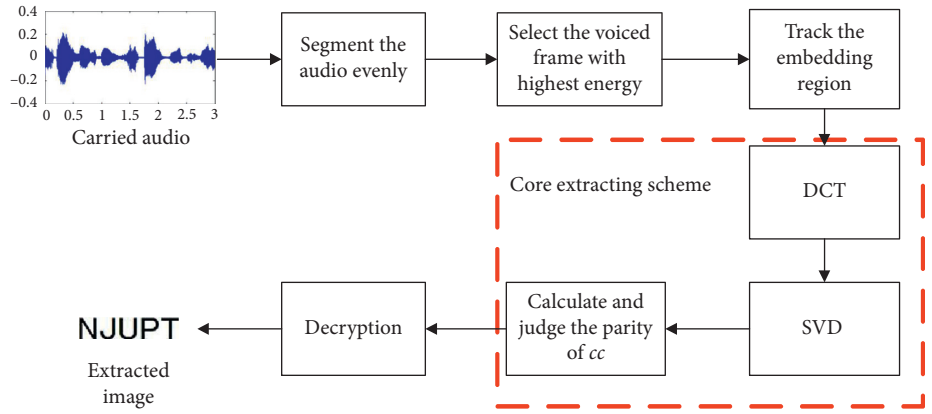

Figure 6: The principle diagram of the extracting algorithm.

where SNR_0 is the lower threshold of transparency. The selected quantization step should not make the algorithm transparency lower than SNR_0 dB. SNR and BER can be defined in formulas.

$$\mathrm{SNR}(A, A') = 10\lg\left(\frac{\sum_{j=1}^{K} A^2}{\sum_{j=1}^{K} (A' - A)^2}\right),$$
$$\mathrm{BER} = \frac{\sum_{i=1}^{L_w} w(i) \oplus w'(i)}{L_w} \times 100\%, \tag{13}$$

where A and A' represent the original audio and the carried audio, respectively, and $w(i)$ and $w(i)'$ denote the original watermark and the extracted watermark. The population consists of C_1 chromosomes, and each chromosome with the length of C_2, which will be encoded by using a binary encoding approach, can be converted into the quantization step. Formula (14) can be used to describe the transformation relationship between each chromosome CH_r and each quantization step a_r $(1 \le r \le C_1)$.

$$a_r = \frac{B2\,D[\mathrm{CH}_r]}{100}, \tag{14}$$

where $B2\,D[\mathrm{CH}_r]$ means converting CH_r from binary to decimal. The detailed process can be described as follows.

(i) Step 1: set the parameters, including the crossover probability p_c and the mutation probability p_m, and then generate an initial population POP_0.

(ii) Step 2: calculate the quantization step a_r according to formula (14) and then execute the embedding algorithm proposed in Section 3.1 after the carried audio is subjected to some attacks.

(iii) Step 3: pick out all qualified chromosomes when $\mathrm{SNR} > \mathrm{SNR}_0$ and then execute the extracting algorithm proposed in Section 3.2.

(iv) Step 4: calculate the fitness value according to formula (12) and obtain the best chromosome with the largest fitness value.

(v) Step 5: perform selection operation by roulette to get the transition population POP_0'.

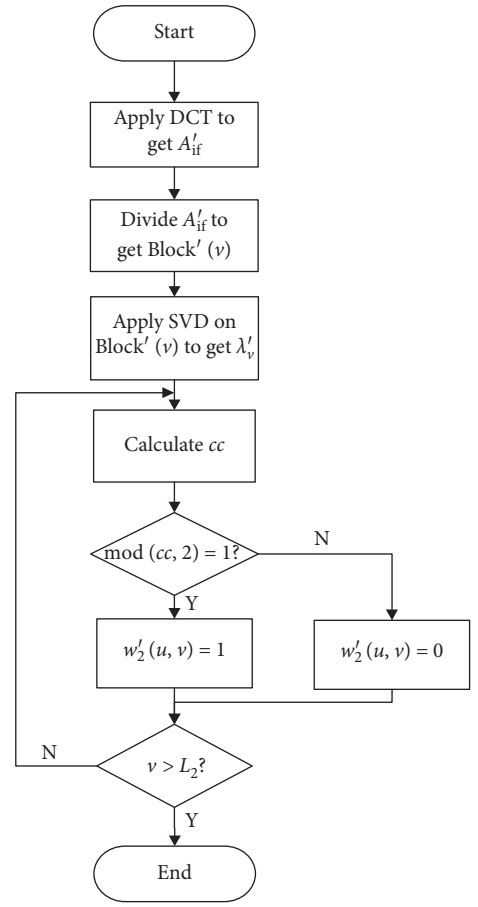

Figure 7: Flowchart of the core extracting scheme.

(vi) Step 6: perform crossover operation on each pair of chromosomes except for the best chromosome to obtain the next transition population POP_0''.

(vii) Step 7: perform mutation operation on each chromosome except for the best chromosome to obtain the next generation population POP_1.

(viii) Step 8: repeat Step 2 to Step 7 until the global optimal chromosome appears.

Table 1: Embedding rule.

Watermark bit	Original parity of cc	Embedding rule	Modified parity of cc
0	Even	$\lambda_y' = a \times cc \pm 0.5a$	Even
0	Odd	$\lambda_y' = a \times cc - 0.5a$	Even
1	Even	$\lambda_y' = a \times cc - 0.5a$	Odd
1	Odd	$\lambda_y' = a \times cc + 0.5a$	Odd

3.4. Measure to Further Improve Robustness. In order to improve the robustness of the algorithm, the same row of the binary watermark can be repeatedly embedded into three voiced frames with the highest energy in A_l. When extracting watermarks, three groups of binary watermarks are extracted from the three voiced frames, respectively, and compared bit by bit to obtain a more accurate group of binary watermarks according to formula (15), where $w_{21}'(u,v)$, $w_{22}'(u,v)$, and $w_{23}'(u,v)$ are the three groups of binary watermarks extracted from three voiced frames, respectively.

$$w_2'(u,v) = \begin{cases} 1, & \sum_{s=1}^{3} w_{2s}'(u,v) > 3/2, \\ 0, & \sum_{s=1}^{3} w_{2s}'(u,v) \le 3/2 \end{cases} \tag{15}$$

4. Performance Evaluation

In this section, the performance of the proposed algorithm will be tested. In order to evaluate the performance of this proposed algorithm, the quality of the audio can be evaluated by three ways, including SNR, the object difference grade (ODG) which is one of the output values obtained from the perceptual evaluation of audio quality (PEAQ), and the mean opinion score (MOS). According to the standard of IFPI, SNR should be greater than 20 dB to make the audio have good transparency. BER can be used to evaluate the algorithm robustness. Generally, small BER means that the algorithm has strong robustness to various attacks. According to the standard of IFPI, the BER of the extracted watermark is no less than 20% when the carried audio is attacked. NC can be used to compare the similarity between the original watermark and the extracted watermark, as shown in formula (16). When NC is close to 1, the original watermark is very similar to the extracted watermark.

$$\mathrm{NC} = \frac{\sum_{i=1}^{L_w} w(i) \times w'(i)}{\sqrt{\sum_{i=1}^{L_w} w(i)^2 \sum_{i=1}^{L_w} w'(i)^2}}. \tag{16}$$

The experimental parameters are as follows. (1) Algorithm parameters: $C_1 = 10$, $C_2 = 8$, $p_c = 0.8$, $p_m = 0.1$, $\mathrm{SNR}_0 = 25$, $N = 4096$, $N_e = 1024$, $L_1 = 64$, $L_2 = 64$, $L_{\mathrm{if}} = 1024$, and $b_0 = 300$. (2) The watermark is a binary image as shown in Figure 8(p), and its size is 128×32. (3) The PEAQ metric was the basic standard which was released by the TSP Lab at McGill University. (4) The tested audio comprises twenty 64-second audio signals, formatted by WAV, sampled at 44100 Hz with 16-bit resolution, including popular songs, classical songs, rock songs, and dialogues.

The detailed experimental environment and software are described as follows: (1) computer system—64-bit Microsoft Windows 10; (2) programming language—Matlab 2016R; (3) software for processing audio signals—Cool Edit Pro V2.1.

4.1. Transparency and Capacity. The payload capacity of this algorithm can be calculated according to the following formula:

$$\mathrm{Cap} = \frac{L_w}{T} = \frac{L_1 \times L_2}{T}, \tag{17}$$

where T is the time length that the audio carries the watermark. In our study, T is equal to 64 seconds, so the payload capacity is $((128 \times 32\mathrm{bit})/64s) = 64$ bps. The average values about the payload capacity (bps), SNR (dB), ODG, MOS of the audio, BER (%), and NC of the extracted watermark are listed in Table 2.

In our test, all the audio signals are processed with the four algorithms mentioned in Table 2, respectively, to obtain four groups of carried audio signals which will be provided to 20 listeners (10 males and 10 females, aged between 18 and 60 years old) in order to get MOS scores. Table 2 shows that this algorithm has good transparency because the average SNR is up to 25.96 dB, ODG is −0.99, and MOS is 4.5 while the payload capacity is 64 bps, which is higher than the standard of IFPI. Most importantly, BER is equal to 0, and NC is equal to 1, which indicates that this algorithm has good robustness when there is no attack, so the extracted watermark image in Figure 8(a) is the same as the original image shown in Figure 8(p). Compared with the algorithms in paper [16] and paper [23], this proposed algorithm has a larger payload capacity and better transparency, and the robustness is stronger than that in paper [23]. Although the payload capacity of this algorithm is not as high as the algorithm in the paper [22], the transparency is more superior. Besides, this proposed algorithm is more robust than other three algorithms, which will be discussed in Section 4.2.

Figure 9 shows the waveform pictures of the carried audio without attack before and after embedding the watermark (we only display an audio clip lasting about 3 seconds to clearly show the details), and their spectrogram pictures are shown in Figure 10. It can be seen that there is no obvious change in the waveform and spectrogram of the audio before and after embedding the watermark, which indicates that this algorithm's transparency is nice.

4.2. Robustness. This section will evaluate the algorithm robustness by BER and NC when resisting against various conventional signal processing operations and synchronous

Figure 8: The extracted images under different processing operations. (a) No attack. (b) MP3 compression with 64 kbps. (c) MP3 compression with 128 kbps. (d) Noise corruption with 20 dB. (e) Noise corruption with 30 dB. (f) Noise corruption with 40 dB. (g) Requantization. (h) Resampling. (i) Echo addition with 50 s. (j) Echo addition with 100 s. (k) Low-pass filtering (4 kHz). (l) Low-pass filtering (8 kHz). (m) Low-pass filtering (12 kHz). (n) Amplitude scaling by 0.8. (o) Amplitude scaling by 1.2. (p) The original image.

Table 2: Experimental results (no attack).

Items	SNR	ODG	MOS	BER (%)	NC	Cap (bps)
Proposed	25.96	−0.99	4.5	0.00	1	64
Paper [16]	21.08	−0.39	4.6	0.00	1	43.07
Paper [22]	21.20	−1.22	4.1	0.00	1	86.13
Paper [23]	24.63	−2.83	3.8	1.28	0.9732	43.07

attacks and compare the experimental results with other algorithms in three related papers.

4.2.1. Conventional Signal Processing Operations. Conventional signal processing operations are the most common attacks encountered by audio in the process of being used and spread, and they may cause damage or even loss of the watermark hidden in the audio, so the watermarking algorithm must have strong robustness to withstand these attacks. These operations mainly include the following types in Table 3.

BER (%) and NC of the extracted watermark are averaged and listed in Table 4 under these signal processing operations. The extracted images whose NC values are closest to the average value are shown in Figure 8. According to the experimental results in Figure 8 and Table 4, this algorithm has strong robustness against conventional signal processing operations, which can be summarized as follows.

When resisting noise corruption with 30 dB and 40 dB, requantization, low-pass filtering with cutoff frequency of 12 kHz, and echo addition with 50 ms, the extracted images are almost the same with the original image. BER values are equal to 0, and NC values are equal to 1, which indicates that the proposed algorithm has excellent robustness against these attacks.

When resisting MP3 compression, noise corruption with 20 dB, low-pass filtering with cutoff frequency of 8 kHz, resampling, echo addition with 50 ms, and amplitude scaling, the extracted images are similar to the original image. BER values are below 1.28%, and NC values are above 0.9740, so the proposed algorithm has good robustness against these attacks.

When resisting low-pass filtering with cutoff frequency of 4 kHz, the former half of the extracted watermark image is very clear, while another half is completely blurred, NC is 0.7458, and BER is 25.64%, as shown in Figure 8(k). The reason for this phenomenon is mainly related to the algorithm parameters, including the length N_e of the data by DCT, the region $[b_0 + 1, b_0 + L_{\text{if}}]$ for embedding the watermark, and the sampling rate f_s of the audio. In our experiment, sampling rate is 44.1 kHz, $N_e = 4096$, $b_0 = 300$, and $L_{\text{if}} = 1024$, so the embedding frequency range $[f_1, f_2]$ can be calculated as follows.

$$f_1 = \frac{(b_0 + 1) \times f_h}{N_e} = \frac{(300 + 1) \times 22.05}{4096} = 1.62\,\text{kHz}, \tag{18}$$

$$f_2 = \frac{(b_0 + L_{\text{if}}) \times f_h}{N_e} = \frac{(300 + 1024) \times 22.05}{4096} = 7.13\,\text{kHz}. \tag{19}$$

It can be seen from formulas (18) and (19) that the watermark can be concealed in the frequency range of [1.62, 7.13] kHz in the audio, so low-pass filtering with a cutoff frequency higher than 7.13 kHz or lower than 1.62 kHz almost has no effect on the watermark.

When resisting low-pass filtering with cutoff frequency of 8 kHz, the extracted watermark is relatively clear, NC is 0.9990, and BER is 0.05%. However, because the upper limit of the embedding region is 7.13 kHz, which is very close to the cutoff frequency (8 kHz) of the filter, and the low-pass filter has 3 dB amplitude attenuation near the cutoff frequency, there are still a few noise points in the extracted watermark image shown in Figure 8(l). When resisting low-pass filtering with cutoff frequency of 12 kHz, the extracted watermark is the same as the original image, as shown in Figure 8(m), NC is equal to 1, and BER is equal to 0. When

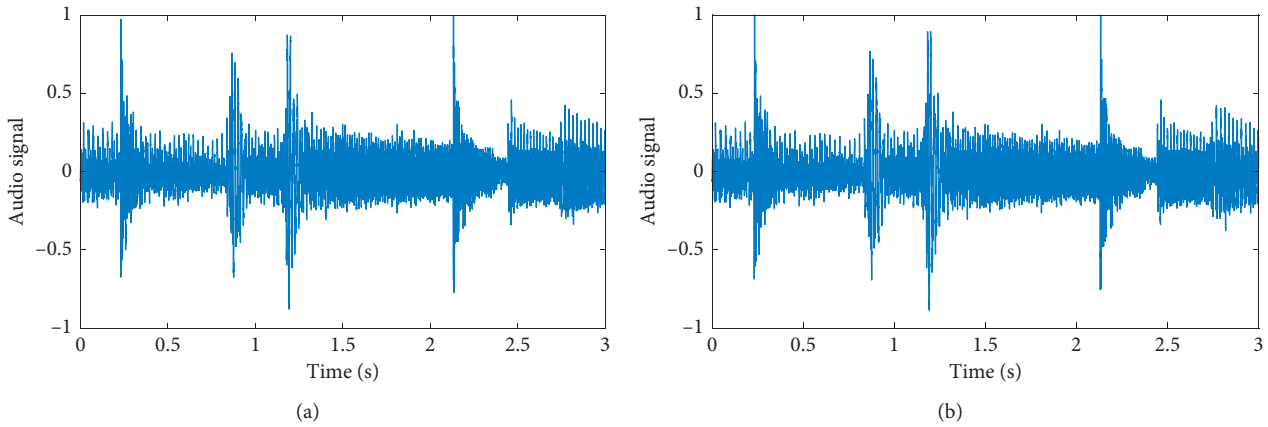

Figure 9: Waveform comparison before and after embedding the watermark. (a) The original audio. (b) The carried audio.

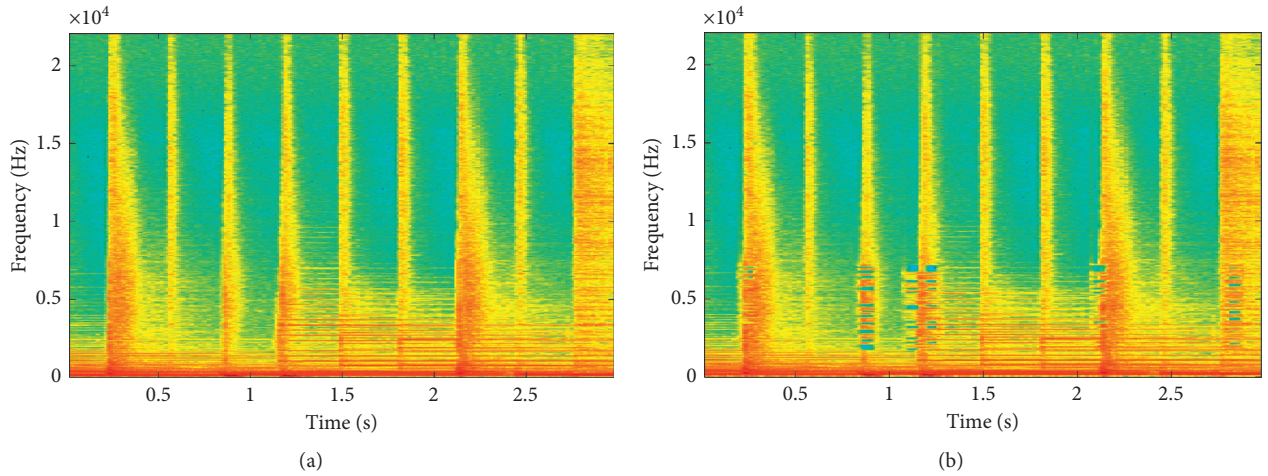

Figure 10: Spectrogram comparison before and after embedding the watermark. (a) The original audio. (b) The carried audio.

Table 3: Attack types and specifications.

Item	Attack type	Description
A	MP3 compression	Applying MP3 compression with 64 kbps and 128 kbps.
B	Noise corruption	Adding Gaussian noise with 20 dB, 30 dB, and 40 dB.
C	Requantization	Quantizing the carried audio with 16 bit-8 bit-16 bit per sample.
D	Resampling	Dropping the sampling rate with 44.1 kHz-22.05 kHz-44.1 kHz.
E	Echo addition	Adding an echo with a delay of 50 ms and 100 ms.
F	Low-pass filtering	Applying low-pass filter with 4 kHz, 8 kHz, and 12 kHz.
G	Amplitude scaling	Scaling the carried audio's amplitude by 0.8 and 1.2.

the audio is subjected to low-pass filtering with cutoff frequency of 4 kHz, the frequency components above 4 kHz in the audio will be removed, so the watermark in the frequency range of [1.62, 4] kHz can be extracted (the former half of the image in Figure 8(k) is very clear), while the watermark in the frequency range of [4, 7.13] kHz cannot be extracted (another half in Figure 8(k) is completely blurred). In practical application, the cutoff frequency of the low-pass filter and the frequency range of the embedding region should be staggered by adjusting the algorithm parameters to prevent watermarks from being damaged.

4.2.2. Synchronous Attack. Synchronization attack is the most challenging type in the research of robust watermarking algorithm.

In Table 5, there are four kinds of synchronization attacks with different strengths for testing the robustness, including TSM, PSM, jittering, and random cropping. After the audio is subjected to the above synchronization attacks, BER (%) and NC values of the extracted watermark are, respectively, averaged and listed in Tables 6–9. The extracted images whose NC values are closest to the average value are shown in Figures 11–14.

Table 4: Average BER (%) and NC of the extracted watermark under different processing operations.

Item	Attack	BER	NC	Item	Attack	BER	NC
A	64 kbps	1.28	0.9740	E	50 ms	0.00	1
	128 kbps	0.79	0.9854		100 ms	0.08	0.9985
B	20 dB	0.07	0.9987	F	4 kHz	25.64	0.7458
	30 dB	0.01	1		8 kHz	0.05	0.9990
	40 dB	0.00	1		12 kHz	0.00	1
C	Down	0.00	1	G	Reduce	0.72	0.9932
D	Down	0.70	0.9872		Enlarge	0.91	0.9876

Table 5: Attack types and specifications.

Item	Attack type	Description
H	TSM	Apply TSM from −10% to 10% on the audio, respectively.
I	PSM	Apply PSM from −5% to 5% on the audio, respectively.
J	Jittering	Delete or add one sample every some samples in the audio.
K	Random cropping	Randomly cut out several samples in the different parts.

Table 6: Average BER (%) and NC values of the extracted watermark under TSM.

Strength (%)	BER	NC
−30	8.79	0.9232
−20	7.64	0.9088
−15	3.74	0.9225
−9	3.61	0.9531
−7	4.05	0.9842
−5	3.34	0.9901
−4	2.91	0.9878
−3	3.08	0.9866
−2	2.51	0.9853
−1	2.73	0.9876
+1	2.34	0.9867
+2	2.56	0.9821
+3	2.71	0.9821
+4	3.61	0.9790
+5	2.69	0.9911
+7	3.47	0.9759
+9	3.32	0.9794
+15	9.13	0.8851
+20	12.04	0.8490
+30	15.31	0.8267

(1) TSM. Table 6 shows the average BER (%) and NC of the extracted watermark under TSM with different strengths from −30% to +30%. It can be seen from the experimental results in Table 6 and Figure 11 that this algorithm has excellent robustness for overcoming TSM attacks with different strengths. BER values all are below 15.31%, which is far superior to the standard of IFPI. NC values all are above 0.8267, so the extracted images all can distinguish its content in them.

(2) PSM. When the audio is subjected to PSM, its playing time will not change, but the position and shape of the voiced frame will change slightly. Table 7 shows the average BER (%) and NC of the extracted watermark under PSM with different strengths from −5% to 5%. Although the extracted images are not very clear in Figure 12, their content can still be distinguished.

(3) Jittering. Table 8 shows the average BER (%) and NC of the extracted watermark under jittering with different strengths from 1/100000 to 1/500. The extracted images are shown in Figure 13. As shown in Figure 13(a), under the maximum attack strength (1/500), the extracted image contains more noise points, but its main feature can still be identified, in which NC is 0.9303 and BER is 3.68%. As the attack strength weakens, the extracted images become more and more similar to the original image in Figure 8(p), and BER values become smaller and smaller, so this proposed algorithm has strong robustness against jittering.

(4) Random Cropping. The average BER (%) and NC of the extracted watermark under random cropping with different strengths are shown in Table 9. From the experimental results, the extracted images in Figure 14 all are relatively clear, NC values are above 0.9754, and BER values are below 1.90%, which all show that this proposed algorithm is robust against random cropping.

4.2.3. Comparative Analysis of Robustness. In order to compare the algorithm robustness with the related algorithms in papers [16], [22], and [23], Table 10 lists BER (%) values of these algorithms when resisting signal processing operations and synchronization attacks. From the experimental results in Table 2 and Table 10, the comparative analysis is discussed as follows about these four algorithms. Compared with the algorithm in paper [16], this proposed algorithm has larger payload capacity, higher transparency, and better robustness against synchronization attacks and conventional signal processing operations except for some attacks, such as MP3 compression with 64 kbps, low-pass filtering with cutoff frequency of 4 kHz, and PSM. According to the embedding principle of this algorithm, the frequency band where the watermark is located can be changed by modifying the algorithm parameters in practical application, so this proposed algorithm can overcome the attack from low-pass filtering with cutoff frequency of 4 kHz in fact. In the following comparative analysis with

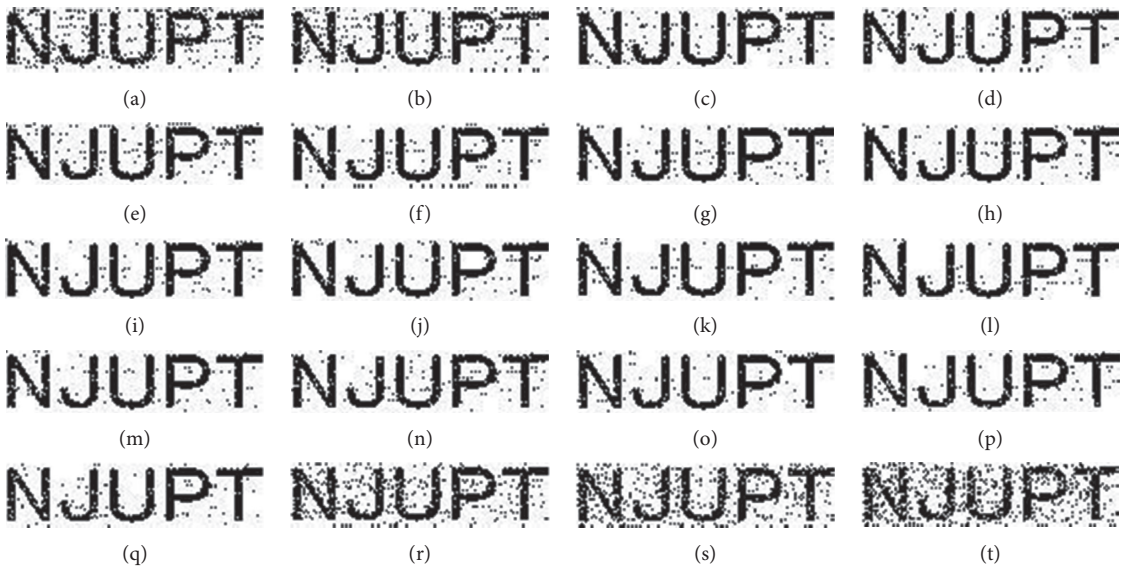

Figure 11: The extracted images under TSM. (a) TSM (−30%). (b) TSM (−20%). (c) TSM (−15%. (d) TSM (−9%). (e) TSM (−7%). (f) TSM (−5%). (g) TSM (−4%). (h) TSM (−3%). (i) TSM (−2%). (j) TSM (−1%). (k) TSM (1%). (l) TSM (2%). (m) TSM (3%). (n) TSM (4%). (o) TSM (5%). (p) TSM (7%). (q) TSM (9%). (r) TSM (15%). (s) TSM (20%). (t) TSM (+30%).

Table 7: Average BER (%) and NC of the extracted information under PSM.

Strength (%)	BER	NC
−5	20.58	0.8680
−4	15.65	0.8923
−3	14.14	0.8988
−2	12.40	0.9333
−1	7.74	0.9112
+1	8.92	0.9156
+2	12.60	0.9297
+3	14.94	0.8977
+4	15.82	0.9037
+5	19.65	0.8772

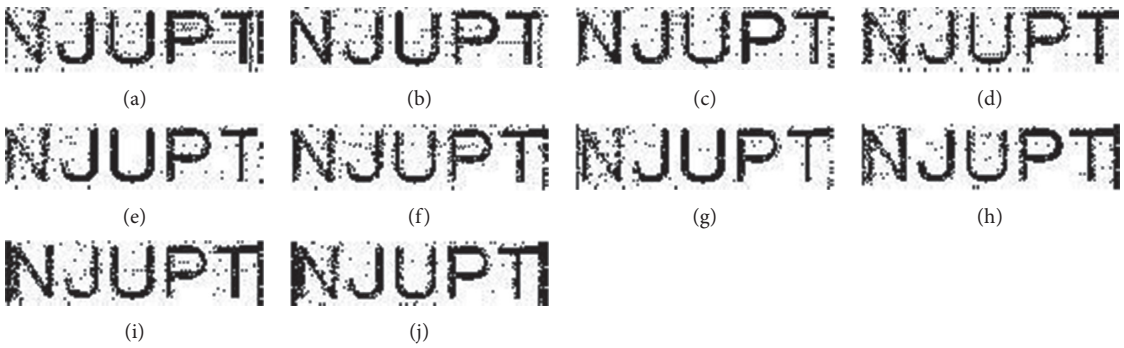

Figure 12: The extracted images under PSM. (a) PSM (−5%). (b) PSM (−4%). (c) PSM (−3%). (d) PSM (−2%). (e) PSM (−1%). (f) PSM (+1%). (g) PSM (+2%). (h) PSM (+3%). (i) PSM (+4%). (j) PSM (+5%).

the other two algorithms, this viewpoint will not be reiterated. The robustness of this proposed algorithm is far superior to the algorithms in paper [22] and paper [23], although the payload capacity is slightly lower than that in paper [22]. Synchronization attacks may change the overall structure of the audio, but it has little effect on the shape of the voiced frame. The proposed ISM in this algorithm can accurately track the position of the largest amplitude in the voiced frame to determine the embedding region where the watermark is located. Therefore, this algorithm has strong robustness against various malicious attacks.

Table 8: Average BER (%) and NC of the extracted information under jittering.

Strength	BER	NC
1/500	3.68	0.9303
1/1000	2.45	0.9628
1/1500	2.42	0.9568
1/2000	1.83	0.9712
1/3000	1.59	0.9718
1/4000	0.97	0.9817
1/5000	0.89	0.9843
1/10000	0.83	0.9877
1/50000	0.67	0.9884
1/100000	0.34	0.9933

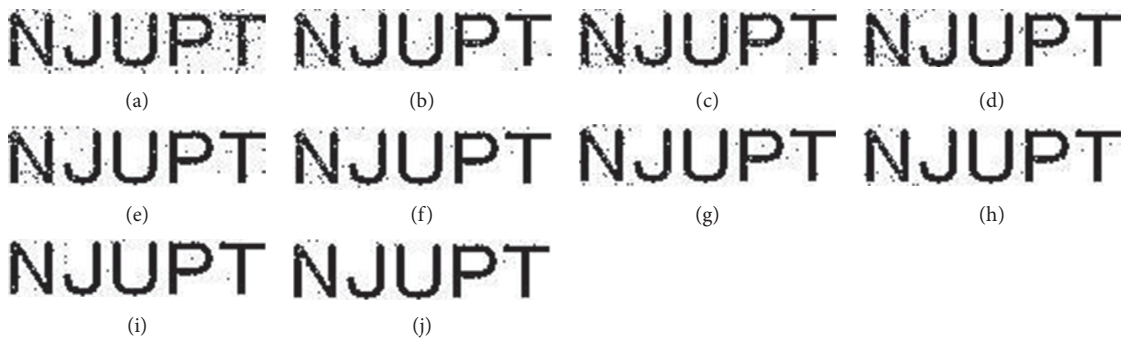

Figure 13: The extracted images under jittering. (a) Jittering (1/500). (b) Jittering (1/1000). (c) Jittering (1/1500). (d) Jittering (1/2000). (e) Jittering (1/3000). (f) Jittering (1/4000). (g) Jittering (1/5000). (h) Jittering (1/10000). (i) Jittering (1/50000). (j) Jittering (1/100000).

Table 9: Average BER (%) and NC of the extracted information under random cropping.

Strength	BER	NC
100 points (front)	0.61	0.9903
100 points (middle)	0.58	0.9912
100 points (back)	0.65	0.9898
200 points (front)	0.81	0.9873
200 points (middle)	0.80	0.9893
200 points (back)	0.84	0.9898
500 points (front)	1.29	0.9798
500 points (middle)	1.05	0.9889
500 points (back)	0.98	0.9897
800 points (front)	1.90	0.9754
800 points (middle)	1.01	0.9863
800 points (back)	1.78	0.9799

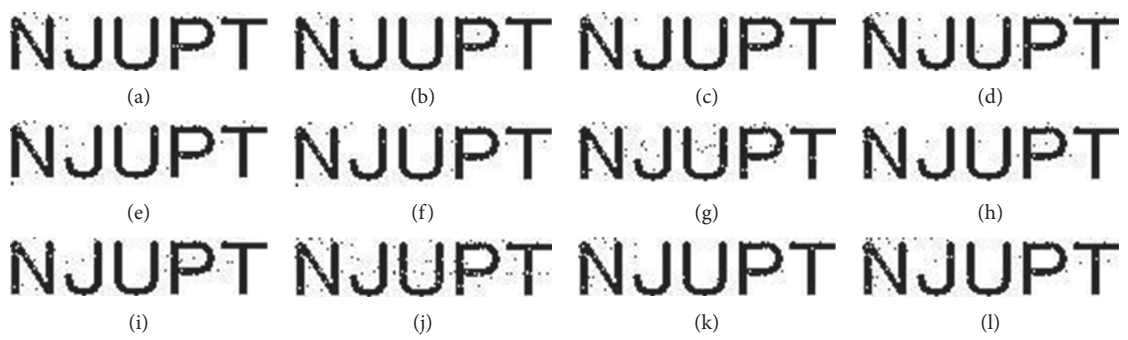

Figure 14: The extracted images under random cropping. (a) 100 points (front). (b) 100 points (middle). (c) 100 points (back). (d) 200 points (front). (e) 200 points (middle). (f) 200 points (back). (g) 500 points (front). (h) 500 points (middle). (i) 500 points (back). (j) 800 points (front). (k) 800 points (middle). (l) 800 points (back).

Table 10: Robustness comparison with other algorithms (BER, %).

Attacks		Proposed	Paper [16]	Paper [22]	Paper [23]
A	64 kbps	1.28	1.93	2.19	26.48
	128 kbps	0.79	0.01	0.04	10.08
B	20 dB	0.07	0.44	16.41	11.47
	30 dB	0.01	0.01	0.00	3.22
	40 dB	0.00	0.00	0.00	1.83
C	Down	0.00	0.01	0.00	2.17
D	Down	0.70	0.00	0.00	2.35
E	50 ms	0.00	0.25	7.03	1.32
	100 ms	0.08	0.19	11.52	0.97
F	4 kHz	25.64	0.00	6.55	50.25
	8 kHz	0.05	0.00	2.02	34.11
	12 kHz	0.00	0.00	0.67	21.16
G	Reduce	0.72	0.00	31.30	1.28
	Enlarge	0.91	0.01	0.39	1.28
H	−5%	3.34	6.45	41.7	6.19
	+5%	2.69	6.51	42.2	6.52
I	−1%	7.74	1.24	38.26	6.75
	+1%	8.92	1.84	37.54	14.59
J	1/500	3.68	0.59	28.15	46.62
	1/1000	2.45	0.33	25.83	37.18
K	200 points	0.82	1.34	7.36	39.28
	100 points	0.62	0.81	6.62	36.73

4.3. Security. The watermark hidden in audio is protected by encryption technology and information hiding technology, so it is necessary to analyse the security of this algorithm from the key space constructed by encryption technology and information hiding technology.

The proposed algorithm uses a triple key $\mathrm{Ch}(x_1, \alpha_0, \delta)$ to encrypt the watermark and seven key parameters (N, N_e, L_1, L_2, b_0, L_{if}, a) to conceal the watermark. $\mathrm{Ch}(x_1, \alpha_0, \delta)$ and a are taken in the real field, so this algorithm has infinite key space in theory. In fact, they are affected by the word length, so their key space is limited. In our test, the computer system is 64-bit, N, N_e, and L_{if} all are 16-bit, and L_1, L_2, and b_0 are 10-bit, so the key space to encrypt the watermark can be calculated as 2^{192}, and the key space to conceal the watermark is 2^{142}. From the above analysis, even if the attacker obtains the principle of the algorithm, as long as these key parameters are not known, it is difficult for the attacker to obtain the watermark.

4.4. Complexity. The complexity of the algorithm is an important index to evaluate the performance of the algorithm. It is usually measured by the computational cost when embedding and extracting the watermark. In our experiment, the average time for extracting the watermark is 0.8544 s and that for embedding the watermark without GA is 1.6246 s. When GA is used to search for the best algorithm parameter, the embedding time is related to the evolution time of GA. It can be seen that GA enables our proposed algorithm to achieve a good balance between transparency and robustness, but it also brings a large computational cost. When embedding the watermark, the average time for searching the embedding region in a voiced frame is 0.037 ms. When extracting the watermark, the average time for tracking the extracting region in a voiced frame is 0.036 ms. It can be seen that our proposed ISM takes up very little computational cost to find the synchronization marks.

5. Conclusions

In our study, it is found that the playing time of the audio will be longer or shorter after being attacked by TSM, but the shape of the voiced frame will not change basically. Therefore, an ISM which can search for the embedding region where the watermark is located is developed, in which it takes the sample point with the largest amplitude in the voiced frame as the synchronization mark. GA is utilized to optimize the key algorithm parameter to balance both transparency and robustness. Combining the "energy concentration" characteristic of DCT and the stability characteristic of SVD, a robust and blind audio watermarking algorithm with ISM and GA is proposed for overcoming malicious synchronization attacks and conventional signal processing operations.

The following measures are taken to improve the algorithm robustness. Firstly, the proposed ISM can accurately track the region where the watermark is located. Even if the structure of the audio changes slightly, the ISM can accurately search this synchronization mark in the voiced frame to track the region that can be used to embed and extract the watermark. Secondly, GA is utilized to optimize the key algorithm parameter to balance both transparency and robustness. Thirdly, the audio will be divided evenly twice to avoid the drift of the embedding region caused by the change of the audio structure. At last, the watermark is repeatedly embedded in three voiced frames to improve the algorithm robustness. Embedding the same watermark in the three voiced frames is equivalent to embedding the watermark with a triple repetition code. The experimental results confirm that this proposed algorithm has excellent robustness in the case that the payload capacity is 64 bps; it can not only withstand conventional signal processing operations but also resist TSM, PSM, jittering, and random cropping. Especially, this algorithm even stands up to TSM with strength from −30% to +30%.

Although the proposed algorithm has excellent robustness when overcoming TSM, jittering, random cropping, and various conventional signal processing operations, the experimental results of this algorithm under PSM are not good enough, mainly because PSM makes the synchronization mark in the voiced frame shift, which leads to the error bits in the extracted watermark. Therefore, the performance of this algorithm is not enough when withstanding some attacks, such as deliberately distorting the peak amplitude points to remove synchronization mark, which will be further studied in our future work. In addition, GA is used to optimize the key algorithm parameter, which is very helpful to balance the transparency and robustness. However, GA needs a long evolution time to search for the optimal algorithm parameters, which greatly increases the computational cost. Based on this, this algorithm is not suitable for the application with strict time requirements. In future research, we will strive to enhance the security and robustness against more types of synchronization attacks.

Data Availability

The data used to support the findings of the study are included within the article and are obtained from public platform.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the High-Level Talent Scientific Research Foundation of Jinling Institute of Technology, China (grant no. jit-b-201918), and the National Natural Science Foundation of China (grant no. 11601202).

References

[1] M. J. Hwang, J. S. Lee, M. S. Lee, and H. G. Kang, "SVD based adaptive QIM watermarking on stereo audio signals," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 45–54, 2017.

[2] Y. Hong and J. Kim, "Autocorrelation modulation-based audio blind watermarking robust against high efficiency advanced audio coding," *Applied Sciences*, vol. 9, no. 14, pp. 1–17, 2019.

[3] B. Lei, I. Yann Soon, F. Zhou, Z. Li, and H. Lei, "A robust audio watermarking scheme based on lifting wavelet transform and singular value decomposition," *Signal Processing*, vol. 92, no. 9, pp. 1985–2001, 2012.

[4] Z. Zhang, M. Zhang, and L. Wang, "Reversible image watermarking algorithm based on quadratic difference expansion," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–8, 2020.

[5] A. Merrad and S. Saadi, "Blind speech watermarking using hybrid scheme based on DWT/DCT and sub-sampling," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 27589–27615, 2018.

[6] G. Hua, J. Huang, Y. Q. Shi, J. Goh, and V. L. L. Thing, "Twenty years of digital audio watermarking-A comprehensive review," *Signal Processing*, vol. 128, pp. 222–242, 2016.

[7] W. Jiang, X. Huang, and Y. Quan, "Audio watermarking algorithm against synchronization attacks using global characteristics and adaptive frame division," *Signal Processing*, vol. 162, pp. 153–160, 2019.

[8] Q. Qian, H. Wang, X. Sun, Y. Cui, H. Wang, and C. Shi, "Speech authentication and content recovery scheme for security communication and storage," *Telecommunication Systems*, vol. 67, no. 4, pp. 635–649, 2018.

[9] M. A. Nematollahi, C. Vorakulpipat, H. Gamboa-Rosales, F. J. Martinez-Ruiz, and J. I. De la Rosa-Vargas, "Digital speech watermarking based on linear predictive analysis and singular value decomposition," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, vol. 87, no. 3, pp. 433–446, 2017.

[10] D. Singh and S. K. Singh, "DWT-SVD and DCT based robust and blind watermarking scheme for copyright protection," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13001–13024, 2017.

[11] H.-T. Hu and L.-Y. Hsu, "Incorporating spectral shaping filtering into DWT-based vector modulation to improve blind audio watermarking," *Wireless Personal Communications*, vol. 94, no. 2, pp. 221–240, 2017.

[12] A. A. Attari and A. A. B. Shirazi, "Robust audio watermarking algorithm based on DWT using Fibonacci numbers," *Multimedia Tools and Applications*, vol. 77, no. 19, pp. 25607–25627, 2018.

[13] P. K. Dhar and T. Shimamura, "Blind audio watermarking in transform domain based on singular value decomposition and exponential-log operations," *Radioengineering*, vol. 26, no. 2, pp. 552–561, 2017.

[14] Z. Liu, Y. Huang, and J. Huang, "Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1171–1180, 2019.

[15] P. Hu, Z. Yi, D. Peng, and Y. Xiang, "Robust time-spread echo watermarking using characteristics of host signals," *Electronics Letters*, vol. 52, no. 1, pp. 5-6, 2016.

[16] H.-T. Hu, J.-R. Chang, and S.-J. Lin, "Synchronous blind audio watermarking via shape configuration of sorted LWT coefficient magnitudes," *Signal Processing*, vol. 147, pp. 190–202, 2018.

[17] S. Xiang and J. Huang, "Histogram-based audio watermarking against time-scale modification and cropping attacks," *IEEE Transactions on Multimedia*, vol. 9, no. 7, pp. 1357–1372, 2007.

[18] H.-T. Hu and J.-R. Chang, "Efficient and robust frame-synchronized blind audio watermarking by featuring multilevel DWT and DCT," *Cluster Computing*, vol. 20, no. 1, pp. 805–816, 2017.

[19] X. Y. Wang, Q. L. Shi, S. M. Wang, and H. Y. Yang, "A blind robust digital watermarking using invariant exponent moments," *AEU-International Journal of Electronics and Communications*, vol. 70, no. 4, pp. 416–426, 2016.

[20] X.-C. Yuan, C.-M. Pun, and C. L. Philip Chen, "Robust mel-frequency cepstral coefficients feature detection and dual-tree complex wavelet transform for digital audio watermarking," *Information Sciences*, vol. 298, pp. 159–179, 2015.

[21] X.-G. Wang, P.-P. Niu, H.-Y. Yang, Y. Zhang, and T.-X. Ma, "A robust audio watermarking scheme using higher-order statistics in empirical mode decomposition domain," *Fundamenta Informaticae*, vol. 130, no. 4, pp. 467–490, 2014.

[22] S.-T. Chen, H.-N. Huang, and C.-Y. Hsu, "Wavelet-domain audio watermarking using optimal modification on low-frequency amplitude," *IET Signal Processing*, vol. 9, no. 2, pp. 166–176, 2015.

[23] L. Li and X. Fang, "Audio watermarking robust against playback speed modification," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E94-A, no. 12, pp. 2889–2893, 2011.

[24] Z. H. Liu, D. Luo, J. W. Huang, J. Wang, and C. D. Qi, "Tamper recovery algorithm for digital speech signal based on DWT and DCT," *Multimedia Tools and Applications*, vol. 76, no. 10, pp. 12481–12504, 2017.

[25] M. A. Nematollahi, S. A. R. Al-Haddad, S. Doraisamy, and H. Gamboa-Rosales, "Speaker frame selection for digital speech watermarking," *National Academy Science Letters*, vol. 39, no. 3, pp. 197–201, 2016.