Journal of Computers Vol. 34 No. 2, April 2023, pp. 53-68
doi: 10.53106/199115992023043402005

Research on Copyright Protection Technology Based on MIDI Music Structural Features

De Li, JiaZheng Gong, ZhaoLong Liu, Xun Jin*

Department of Computer Science, Yanbian University,
Yanji, China
leader1223@ybu.edu.cn, 2300191944@qq.com, 1119160747@qq.com, xunjin@ybu.edu.cn

Received 17 May 2022; Revised 20 August 2022; Accepted 19 September 2022

Abstract. Due to the extensive use of Musical Instrument Digital Interface (MIDI) music in education, entertainment and business, copyright infringements of MIDI music are increasing rapidly. The low robustness of existing MIDI music watermarking is the main problem. Therefore, this paper proposes two novel MIDI music watermarking algorithms to comprehensively protect the copyright of MIDI music. We first propose a dual threshold audio watermarking algorithm based on MIDI note time difference and velocity value. It extracts a main melody audio track of MIDI music through a Ho-Kashyap (H-K) algorithm, generates an adaptive em-bedding double threshold for the main melody audio track, and uses a time fluctuation algorithm to embed the watermark information. We also propose a MIDI music watermarking algorithm in frequency domain using Discrete Cosine Transform (DCT). It converts the pitch of MIDI music to the fundamental frequency, obtains the Direct Current (DC) component of the fundamental frequency by DCT transform, combines the watermark information and the DC component to obtain the secret pitch, and takes the difference between the secret pitch and the original pitch as the new watermark information. Finally, a general attack experiment was conducted on the two watermarking algorithms to verify the feasibility and security. In this paper, the mean square error method and the normalized correlation coefficient method are selected to evaluate the quality of MIDI music objectively. An absolute information entropy evaluation model is proposed to evaluate MIDI music quality. Experimental results show that the proposed algorithms can extract effective watermark after MIDI command attacks with a MIDI music cutting rate of more than 88% and tampering rate of less than 5%, which means that the proposed algorithms are robust against attacks.

Keywords: copyright protection, MIDI music, robust watermarking, discrete cosine transform, audio feature

1 Introduction

Musical Instrument Digital Interface (MIDI) protocol was first proposed for interface of unified electronic musical instruments in 1980s. The formulation of the MIDI protocol has made great progress in digital music. Since audio transmission has entered the first boom period [1]. MIDI technology plays an important role in the field of music art because of the advantages of low investment, high efficiency and quality. Especially, in the field of music education, MIDI music is essential for electronic music, music appreciation, and ear training classrooms. With the advancement of multimedia technology and equipment, MIDI music is not only widely used in vocal music creation and music dubbing for movies, but also used in game sound effects, TV station titles and advertising music. Due to the extensive use of MIDI music in education, entertainment and business, infringing activities such as illegal downloading, modification, and dissemination of MIDI music are increasing. They destroy the securities of MIDI music contents and seriously damage the copyrights of content owners. According to statistics, more than 90% of digital music works shared on the Internet are not approved by the original creators. In order to ensure that the legitimate interests of MIDI music owners are not infringed and to ensure the safety of the MIDI content, the copyright of MIDI music needs protection.

The key research problems are summarized as follows:

(1) The MIDI structural features should be analyzed to extract appropriate component for embedding watermark, so that we can embed sufficient capacity of watermark information.

(2) Study the adaptive threshold calculation method, to improve the transparency and concealment of watermark.

(3) A robust MIDI music watermarking technology is necessary for resisting MIDI music attacks.

---

* Corresponding Author

The main contributions of this paper are summarized as follows: According to the characteristics of the MIDI format protocol, we propose two MIDI music watermarking methods. One is an improved watermarking based on spatial domain of MIDI music. The other one is a novel MIDI music watermarking based on DCT domain. Because of using the adaptive threshold in the embedding process, the transparency is significantly im-proved. With the Arnold scrambling operation, we further ensure the security of the watermark. Because of the DC components of DCT coefficients are selected to embed watermark, the performance of the algorithm remains high robustness. The watermark is embedded into the MIDI pitch, thus the we can achieve high concealment. To verify the feasibility of these algorithms, several conventional aggression experiments are carried out. We also analyze the transparency of the algorithms. We use two objective quality evaluation methods for MIDI music quality analysis. An objective quality evaluation model of absolute entropy is proposed. The proposed methods have significant impacts on the copyright protection of MIDI music and research field of audio watermarking.

The remainder of the paper is organized as follows. In Section 2, we review digital audio watermarking technologies. In Section 3, we propose the dual threshold audio watermarking based on MIDI note time difference and velocity value. In Section 4, DCT based MIDI music watermarking algorithm is presented. Experimental results are shown in Section 5. Finally, we conclude with the proposed method in Section 6.

2 Related Work

The word "Water Mark" officially appeared in a literature for the first time in 1993, marking the birth of watermarking technology [2]. Digital watermarking refers to hiding personal signs such as text, trademarks, digital images or audio to prove the ownership between the creator and the carrier. Because of the imperceptibility of watermark, users cannot perceive the existence of the sign from the surface. The watermark is extracted as an evidence by a specific detector. Generally, the watermark is hidden in the original space of the carrier and does not occupy additional space. Digital watermarking technology is an important branch of information hiding. The research of audio watermarking has attracted many researchers [2-14]. Digital audio watermarking technology can achieve the protection and certification of digital copyright by extracting the information embedded in the audio. It can track and identify pirated audio works through the watermark information. At present, there are many researches on waveform digital audio watermarking technologies. Most of the audio watermarking algorithms are achieved in spatial and transform domains. In [5], an audio watermarking method was proposed based on voiceprint of speech signal using wavelet transform. It can only be used for speech audio protection. In [7], an adaptive audio watermarking algorithm in the wavelet domain was proposed based on principal component analysis. It extracts discriminatory features from the waveform audio signal. However, it is not applicable to the copyright protection of MIDI files. Because the storage form and content meaning of a MIDI file are different from waveform audio files. In [8-9], two Least Significant Bit (LSB) based audio steganography methods were proposed to achieve audio watermarking. At present, the audio watermarking algorithms that can be used for MIDI music are LSB algorithms based on force values and ellipsis instructions. However, these algorithms are not robust enough. In [10], an audio watermarking scheme was proposed based on Discrete Cosine Transforms (DCT). It embeds the watermark bits according to the effects of the compression of the embedded audio signal. However, this algorithm cannot be used for embedding watermark into the MIDI music, because the MIDI music files are not compressed files. Therefore, we need to pay more attention to the research of MIDI music watermarking. In [11], an audio watermarking algorithm was proposed for tracing the re-recorded audio source. It uses logarithmic mean of coefficient feature to embed watermark and achieves robustness against re-recording attacks. However, it cannot resist cutting attacks, adding noise attacks and tampering attacks.

Several audio watermarking methods have been proposed based on spatial domain of MIDI music [12-16]. In [12], an audio watermarking algorithm was proposed for MIDI music based on the optical music recognition. However, this algorithm is useful in integrity auditing and synchronized transmission, lack of robustness against attacks. Adli proposed three steganography algorithms for MIDI files [13]. The first one is a LSB velocity value based watermarking algorithm. This algorithm changes the MIDI music velocity according to the auditory masking effect, but it is not robust enough. The second one is a steganography algorithm with system exclusive commands. This algorithm embeds the watermark information into positions of MIDI file attributes, such as titles. However, this algorithm increases sizes of MIDI files. The third one is to omit the commands of MIDI files. If the adjacent commands of control switches in a MIDI file are the same, the latter control command can be omitted. The adjacent repeated commands will be automatically ignored in the sequencer or in the MIDI playback device. A steganography scheme was proposed for MIDI music files based on tempo values in [14]. It uses the tempo values of the first strong beat and first weak beat to select reference values. It embeds secret information into the velocity values, excluding reference values. In [15], a method of hiding the reversible data in a MIDI file is proposed by adjusting the incremental time value. It moves the division value and non-zero incremental time value simultaneously when shifting the MIDI file instruction to the left. The secret information is embedded into the non-zero incremental time value generated by the left shift. In [16], a MIDI file steganography algorithm is proposed based on duration fluctuations. Instead of using the MIDI message itself to express embedded information, it uses fluctuation which is added by a player or producer. When the original MIDI file is distributed over the Internet, this method cannot protect the copyright of the MIDI file. Different from the existing methods, in this paper, the watermark is embedded through the time difference, velocity value and pitch. With the time difference and the velocity, the adaptive threshold is generated. We transform the pitch to frequency domain using DCT and the DC component is used for embedding.

3 Dual Threshold Audio Watermarking Based on MIDI Note Time Difference and Velocity Value

3.1 Preprocessing of Carrier Signal and Watermark Signal

(1) Carrier signal preprocessing

Generally, the extraction of the main melody is to estimate the pitch sequence of the main melody from the music signal, which is mainly based on the saliency and timing continuity of the main melody component in the music. The traditional main melody extraction methods are divided into multi-pitch estimation methods and melody contour construction methods. According to music theory analysis and MIDI music file structure characteristics, the difference between MIDI music main melody track and sub-melody track is mainly reflected in the balance of left and right channels, volume, length, velocity, pitch and interval. The first five feature quantities can be extracted directly from the MIDI file. Several music features used in this research are as follows. Left and right channel balance $x_1 = 1 - \frac{|Balance_i - 64|}{64}$, which characterize the proportion of the volume of the notes of the track in the left and right channels. Master volume $x_2 = \frac{Volume_i}{127}$ is the sound amplitude. Average value $x_3 = \frac{1}{n}\sum_{i=1}^{n} Duration_i$ of the duration of the note depends on the duration of the vibration time. Average strength $x_4 = \frac{1}{n}\sum_{i=1}^{n}\frac{Velovity_i}{127}$ characterizes the strength of the sound. It is used to characterize the energy of each track. Average pitch $x_5 = \frac{1}{n}\sum_{i=1}^{n}\frac{Pitch_i}{127}$ refers to various sounds of different heights. The height of the sound is determined by the vibration frequency of the pronunciation body. Interval ratio which refers to the absolute value of the difference between two adjacent pitches. Melody intervals are mainly concentrated from 0 to 6 degrees, but intervals above octave rarely occur. The interval calculation formula is defined as follows:

$$Interval_j = \begin{cases} |Pitch_{j-1} - Pitch_j| & \text{if } |Pitch_{j-1} - Pitch_j| < 25 \\ 25 & \text{if } |Pitch_{j-1} - Pitch_j\ | \geq 25 \end{cases}, \tag{1}$$

The $Interval_j$ is the absolute value of two adjacent pitches, and $Pitch_j$ is the pitch of MIDI music. The formula for calculating the interval ratio of audio tracks from 0 to 6 degrees is defined as follows:

$$x_6 = \frac{ToneM_i}{Tone_i} = \frac{\sum_{i=0}^{6} Count_i}{\sum_{i=0}^{25} Count_i}, \tag{2}$$

The x_6 is the interval ratio, and $Count_i$ is the number of measurement of i degrees.

We use the Ho-Kashyap (H-K) algorithm to extract the main melody of MIDI. Generate a 6-dimensional feature vector $x = (x_1, x_2, x_3, x_4, x_5, x_6)^T$ and the optimal hyperplane $d(x) = w_0 x + w_7$, $w_0 = (w_1, w_2, w_3, w_4, w_5, x_6)$. Normalize the feature symbols to obtain the augmented matrix X. Define N-dimensional co-vector $b > 0$ and calculate the minimum value of the square error standard function as the basis for judging the main melody track. The calculation formula is defined as follows:

$$w_0'X \geq b > 0 \text{，} \tag{3}$$

$$J(w,b) = \|w_0'X - b\| = \sum_{i=1}^{n}(w_i x_i - b_i)^2 \text{，} \tag{4}$$

(2) Watermark signal preprocessing

The original watermark image is converted to a binary sequence image with 0 and 1. Select an appropriate parameters μ and an initial value x_0 as the key to generate a one-dimensional chaotic sequence $X = \{x_1, x_2, \ldots, x_k\}$, where the value of k needs to be equal to the size of the watermark image. The purpose of using one-dimensional Logistic chaotic mapping is to destroy the pixel correlation of the watermark image and improve the security of the watermark. The watermark image after Logistic chaotic encryption is shown as Fig. 1.

(a) Original watermark (b) Logistic scramble of the watermark

Fig. 1. The original watermark and the logistic scramble of the watermark

3.2 Watermark Embedding Algorithm

Generally, MIDI music watermarking algorithms include LSB algorithm, omitting command algorithm and system exclusive command algorithm. LSB algorithm mainly embeds watermark information into a single MIDI command. It has a large capacity but low robustness. The omission instruction algorithm has a low embedding capacity, and the embedding capacity changes with the size of the MIDI file. Moreover, it is a fragile algorithm. The system exclusive instruction algorithm steganography in the MIDI file to represent the music information in the title, this method can embed a lot of watermark information, but will increase the redundant MIDI file information. Because of the MIDI music file belongs to the protocol communication level, unlike the compressed music, MIDI music stores the instructions of the sound. The most of the waveform audio watermarking algorithms are not suitable for MIDI music watermarking. To achieve a robust MIDI music watermarking, the properties of MIDI music structure need to be studied.

In this paper, we select a MIDI music with size of 19.1Kb as the carrier audio. According to the embedding capacity of the carrier, a binary image of 30 × 30 size is selected as the watermark image. The structure of MIDI music is composed of time difference, state, pitch and velocity. We first use the H-K algorithm to extract the main melody of MIDI music. Then we analyze the main melody track to extract the time difference and velocity value parameters. Calculate the embedding threshold through statistical calculation. We combine the MIDI music time fluctuation algorithm and the velocity value LSB algorithm to embed the watermark in two parameters at the same time. It can improve the capacity of the watermark and the ability to resist external attacks.

The watermark embedding process is shown as Fig. 2. It considers the case of multiple melody tracks. The watermark is embedded without discriminating the main melody track. The embedding process is described as follows:

Step 1: We use the H-K algorithm to extract the main melody audio track of MIDI music.

Step 2: The time difference and velocity values in the extracted melody track are obtained using $(wa[i]\&0xf0) = 0x90$ and stored in the Wa_n array.

Step 3: The parameter values of $max\,1$ and $max\,2$ denote the number of occurrences in Wa_n sequence, and generate the embedding threshold $threshold$. The main data structure of MIDI music is composed of time difference, type, note and velocity. To generate the adaptive threshold, the time difference and velocity value in the main melody are extracted. The value with the most occurrences of the parameter and the second value are counted as $max\,1$ and $max\,2$. The n adaptive threshold is generated as follows:

$$threshold = \begin{cases} max+2 & \text{if (max)mod2=0} \\ max+1 & \text{if (max)mod2=1} \end{cases}, \tag{5}$$

where the max is the larger one of $max\,1$ and $max\,2$.

Filter the parameters of $Wa_n > threshold$ and store them in the A array. Convert A into a two-dimensional matrix. The matrix order is $\lfloor sqrt(A_{length}) \rfloor$. Perform Arnold scrambling operation on the A matrix, and record it as B.

Step 4: Select an appropriate parameter μ and an initial value x_0 as the key, and perform Logistic processing on the watermark image.

Step 5: The time difference distribution of MIDI files is regular, which is roughly distributed around several fixed values. The time fluctuation algorithm is used to embed the watermark information into the B array. The specific embedding method is to generate the fixed value portion of the time fluctuation, which is defined as $average$ as follows:

$$average = \begin{cases} \frac{max1+max2}{2}+0 & \text{if } (\frac{max1+max2}{2})\text{mod2=0} \\ \frac{max1+max2}{2}+1 & \text{if } (\frac{max1+max2}{2})\text{mod2=1} \end{cases}, \tag{6}$$

The calculation formula of time fluctuation is as follows:

$$d_i = Wa_i - average \;\; \text{(i=1,2,...,n)}, \tag{7}$$

$$Wa_i = average + bitget(d, 1, wm(1,i)) \;\; \text{(i=1,2,...,n)}, \tag{8}$$

Step 6: Use the Arnold inverse transform to restore the B array containing the watermark. Use the parameter values in the restored B array to replace the parameter values in the original main melody track. Combine the watermarked MIDI main melody track with the sub-melody track to achieve watermarked MIDI music.

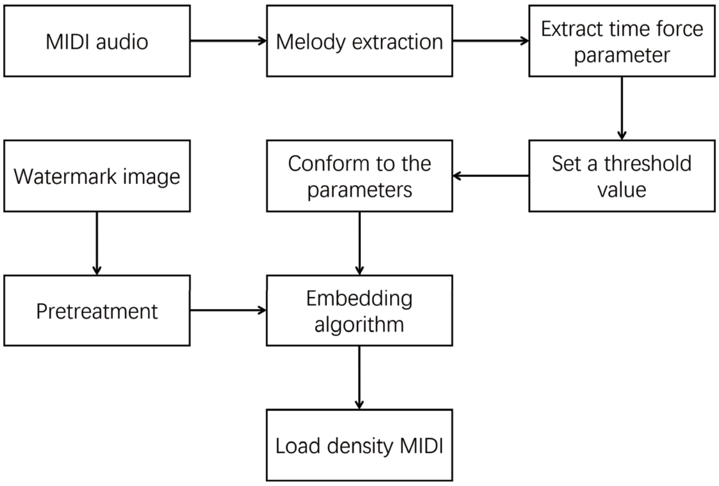


Fig. 2. Watermark embedding flow chart

3.3 Watermark Extraction Algorithm

The watermark extraction process is the inverse process of watermark embedding as shown in Fig. 3. The algorithm is described as follows.

Step 1: Use the H-K algorithm to extract the main melody audio track of MIDI music.

Step 2: Use $(wa[i]\&0xf0) = 0x90$ to extract the time difference and velocity value of the MIDI main melody track and store it in the Wa_n array.

Step 3: Count the parameter values $max\,1$ and $max\,2$ in the Wa_n array with the most and the second most occurrences, and calculate the average and generate the embedding threshold using equation (5) and (6).

Step 4: Filter the $Wa_n > threshold$ parameters and store them in the A array, convert the A array into a two-dimensional matrix, perform Arnold scrambling operation on the matrix, and record it as B;

Step 5: Use the extraction algorithm to extract the watermark information as follows:

$$wm_i = bitget(d,1)wm_i = bitget(d,1)\text{，} \tag{9}$$

where the d is obtained by equation (7).

Step 6: Convert the watermark sequence into a two-dimensional matrix. Decrypt the two-dimensional matrix with the key x_0 and μ_0 using the inverse Logistic model, and restore the watermark image.

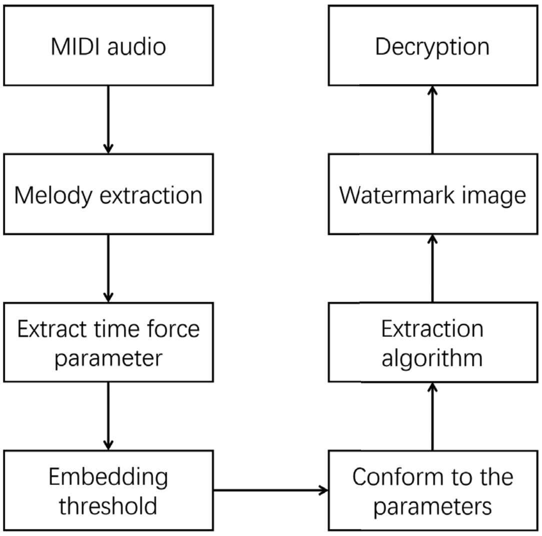


Fig. 3. Watermarking extraction flow chart

4 MIDI Music Watermarking Algorithm Based on DCT Domain

4.1 Preprocessing of Carrier Signal and Watermark Signal

(1) Carrier signal preprocessing

First, the H-K main melody track extraction algorithm and the dual threshold audio watermarking algorithm are used to extract the main melody of carrier MIDI music based on MIDI note time difference and velocity value. The sound is generated by the vibration of the object. There are four basic characteristics in the sound: loudness, tone, timbre and tone. The higher the frequency is, the higher the pitch is. Therefore, the pitch of MIDI instruction can be converted into the fundamental frequency of sound with the next formula.

$$f = 440e^{(m-69)*ln(2)/12} , \tag{10}$$

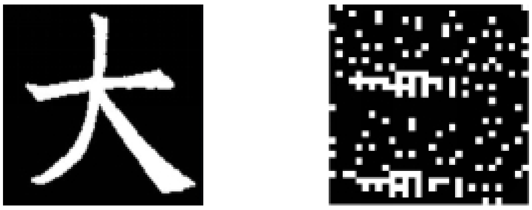
where m represents pitch.

The frequency of the sound that the human can hear is between 20-20000 Hz, thus the fundamental frequency can be normalized as follows:

$$f = f/20000 , \tag{11}$$

(2) Watermark signal preprocessing

To destroy the pixel correlation of the original watermark image and to improve the security of the watermark image, the binary watermark image is scrambled with a key string k_3k_3 using the Arnold transform. The watermark image after scrambling is shown as Fig. 4. The scrambled watermark image is subjected to dimension reduction processing to generate a one-dimensional sequence $Wm_n = \{Wm_1, Wm_2, \ldots, Wm_i\}$ $(0 < i \leq n)$.



(a) The original watermark (b) Arnold scramble of the watermark

Fig. 4. The original watermark and the Arnold scramble of the watermark

4.2 Watermark Embedding Algorithm

The MIDI music time domain watermarking algorithm has low time complexity, but it is poor to resist to external attacks. The transform domain watermarking algorithm has better robustness. We extract the main melody track and parse out the pitch attribute of the main melody track according to the methods proposed in Section 2. Convert the pitch into the fundamental frequency based on the principle of object pronunciation. The watermark information is transformed to the MIDI transform domain. The DC coefficient after DCT transformation is more obvious than Alternating Current (AC) coefficient. Therefore, the DCT transform of pitch fundamental frequency property is used to separate the DC coefficient. The DC coefficient is combined with the watermark information to generate a new sequence to improve watermark security. At the same time, the time fluctuation algorithm is applied to the MIDI strength value for watermark embedding. The watermark embedding process is shown as Fig. 5. It is described as follows:

Step 1: Use the H-K algorithm to extract the main melody audio track of MIDI music.

Step 2: Extract the pitch of the MIDI main melody track and store it in the Wa_n array.

Step 3: Convert the pitch Wa_n to the fundamental frequency f, and normalize the fundamental frequency.

Step 4: Since the DCT transform compresses the energy in the data block into a small number of low-frequency coefficients, it has better energy compressibility. The absolute value of the DC coefficient is usually much larger than that of the AC coefficient. The $1 \times n$ dimension normalized fundamental frequency matrix is expanded to a $m \times n$ matrix, where m is the number of custom expanded matrix rows. The expanded $m \times n$ order fundamental frequency matrix is transformed to extract the maximum DC component of each dimension as follows:

$$F(u,v) = \frac{2}{N} C(u)(v) \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} f(i,j) \times cos[\frac{\pi(2i+1)u}{2N}]\, cos[\frac{\pi(2j+1)v}{2N}] , \tag{12}$$

The DC component after the DCT transform of each dimension is:

$$d_n(i) = \{i = 1,2,3,\ldots,n\}, \tag{13}$$

Step 5: Use the Arnold model to scramble the watermark image to enhance the security of the watermark, and convert the scrambled watermark to a one-dimensional sequence.

Step 6: The embedding strength of the watermark is defined as the quantization step size Δ. If the Δ is too large, it will affect the transparency. If it is too low, it will reduce robustness. Embed the watermark information into the DC component. The watermark is embedded according to the pre-processed Wm_n as follows:

$$d_k(i) = \begin{cases} (Q+1)\times\Delta; & \text{if Qmod2=0} \\ Q\times\Delta; & \text{if Qmod2=1} \end{cases} \quad \text{if } \mathrm{wm}_i = 1(0 < i < n), \tag{14}$$

$$d_k(i) = \begin{cases} Q\times\Delta; & \text{if Qmod2=0} \\ (Q+1)\times\Delta; & \text{if Qmod2=1} \end{cases} \quad \text{if } \mathrm{wm}_i = 0(0 < i < n), \tag{15}$$

where $Q = round(d_k(i)/\Delta)$.

Step 7: Inverse Discrete Cosine Transform (IDCT) is applied to the DC component after embedding the watermark. Multiply the obtained coefficient by 20000 to perform inverse normalization operation to obtain the watermarked fundamental frequency sequence. Then convert the fundamental frequency sequence into MIDI pitch. The conversion method is defined as follows:

$$m = \frac{12*ln(\frac{f}{440})}{ln(2)} + 49, \tag{16}$$

Step 8: The difference between the original pitch and the embedded pitch is defined as $d_i = H_{watermarked} - H_{orignal}$. Use the LSB algorithm to embed the d_i into the MIDI velocity parameters as follows:

$$Wa_i = bitset(Wa_i, 1, wm(d,i)) \;\; \text{(i=1,2,...,n)}, \tag{17}$$

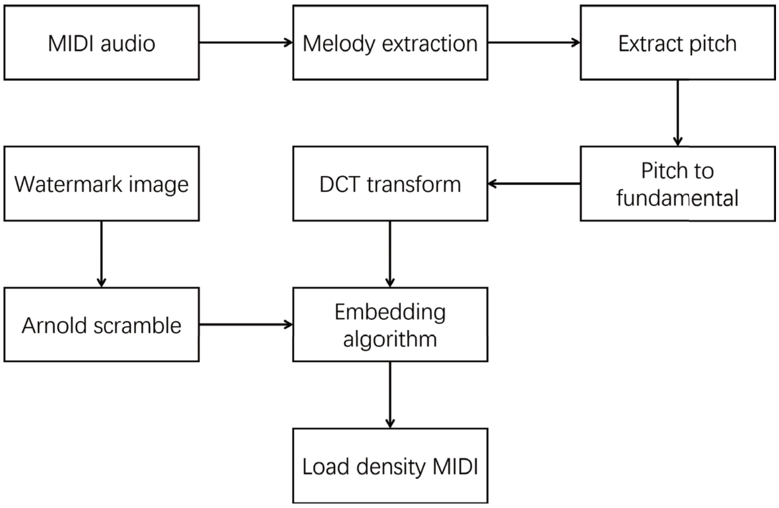


Fig. 5. Watermark embedding flow chart

4.3 Watermark Extraction Algorithm

The watermark extraction process is the inverse process of watermark embedding as shown in Fig. 6. The algorithm is described as follows.

Step 1: Use the H-K algorithm to extract the main melody of MIDI music.

Step 2: Extract the pitch of the main melody of MIDI and store it in the Wa_n array, and store the velocity value in Wb_n.

Step 3: Use the LSB algorithm to extract the watermark wm_n and form the sum with the pitch Wa_n to obtain the watermarked pitch sequence Wc_n.

$$wm_i = bitget(Wb_i, 1) \ \ (\text{i=1,2,...,n}), \tag{18}$$

$$Wc_i = Wa_i + wm_i \ \ (\text{i=1,2,...,n}), \tag{19}$$

Step 4: Convert the pitch in Wc_n to the fundamental frequency, and normalize the fundamental frequency as follows:

$$f = 440e^{(Wc_i-69)*ln(2)/12}/20000 \ \ (\text{i=1,2,...,n}), \tag{20}$$

Step 5: The fundamental frequency is expanded into a matrix of $m \times n$, where m is the number of expanded lines while embedding. Perform DCT transform on the expanded fundamental frequency matrix to extract the DC component matrix $d_n(i) = \{i = 1,2,3,\dots,n\}$.

Step 6: Extract the watermark with the quantization step size Δ and the d_n as follows:

$$Wm_i = \begin{cases} 0, & \text{if} \ \ (\text{round}(d_n(i)/\Delta))\%2=0 \\ 1, & \text{if} \ \ (\text{round}(d_n(i)/\Delta))\%2=1 \end{cases}, \tag{21}$$

Step 7: Convert Wm into a matrix of $m \times m$ size and use the Arnold model to scramble and decrypt to obtain the watermark image.

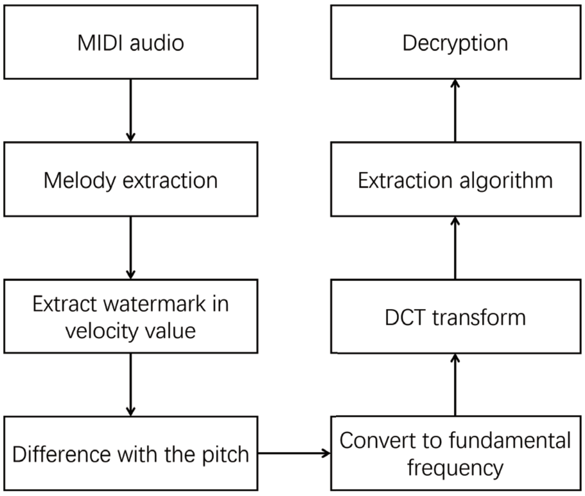


Fig. 6. Watermark extraction flow chart

5 Experimental Results and Analysis

5.1 Attack Experiment

Generally, audio attack methods include resampling, low pass filtering attacks, compression attacks, and shear attacks in formats of wav and mp3. Most of them can completely destroy MIDI music, thus they are not suitable for MIDI music attacks. MIDI music records instructions for sound playback and does not have a waveform structure. The MIDI music command structure has a high degree of fragility. The MIDI music modified by different sequence software can cause very different command data. In this paper, we use a Cubase sequencer to attack MIDI music. The experimental platform is Matlab. The attack methods include cutting off some audio tracks, adjusting audio track order, adding noises and tampering with some commands. A MIDI music with size of 19.1Kb is used for the attack experiments. The audio used for the attack experiment contains 64 sets of watermark images, which means the 64 sets of effective watermark images can be extracted from the watermarked MIDI music without attacks. Fig. 7(a) shows the audio signal without watermark. Fig. 7(b) shows the embedded audio signal. There is nearly no difference between them during auditory test, because of combining the watermarked MIDI main melody track with the sub-melody track to achieve watermarked MIDI music. Fig. 8(a) shows the extracted watermark image. Fig. 8(b) shows the decrypted watermark image. The embedded watermark is shown as in Fig. 1. The bit error rate between the original watermark and the decrypted watermark is 0% without attack.

(a) Audio signal without watermark (b) Audio signal with watermark

Fig. 7. Audio signals with and without watermark

(a) Extracted watermark image (b) Decrypted watermark image

Fig. 8. Extracted and decrypted watermark images without attack

(1) Cutting attack: The cutting rate refers to the ratio of the MIDI music size before and after cutting. We use Cubase sequencer to properly cut the watermarked MIDI music. The MIDI music after cutting is shown in Fig. 9. Watermark extraction was performed on the clipped MIDI music to detect the anti-clipping performance. The extracted watermark is shown in Fig. 10. The experimental results are shown in Table 1. The experimental results show that when the cutting rate is 88%, the dual threshold audio watermarking algorithm can still extract 4 sets of effective watermark images. Because of the DC coefficient is combined with the watermark information, the DCT based audio watermarking algorithm can still extract 5 effective watermark images. The two algorithms can extract effective watermark information after the cutting attacks, which indicates that the proposed algorithms can resist the cutting attacks. When the strength of the cutting rate highly increased, the audio will be damaged seriously, thus the watermark images are hard to be extracted.

Fig. 9. Audio signal after cutting attack

(a) Extracted watermark image (b) Decrypted watermark image

Fig. 10. Extracted and decrypted watermark images after cutting attack

Table 1. Watermark results after cutting attack

Cutting ratio	97%	88%	75%	43%
Audio size after cutting (kb)	18.6	16.8	14.3	8.13
The first method	12	4	0	0
The second method	11	5	0	0

(2) Adjusting audio track order attack: MIDI music has sixteen channels. They are divided into different audio tracks. Each audio track represents a different pronunciation instrument. The audio track order adjustment attack will change the MIDI music main and auxiliary melody pronunciation instruments. Adjusting the MIDI music track order will cause the MIDI music tone to change. The MIDI music map after adjusting the track order is shown in Fig. 11. Watermark extraction of MIDI music after adjusting the order of audio tracks is carried out to detect the anti-attack ability of watermark. Because of using the H-K algorithm to extract the main melody audio track of MIDI music, the experimental results show that the two proposed algorithms can extract 15 effective watermark images. Which means that the proposed algorithms can resist the attack of adjusting the sound track order.

(a) Audio signal before adjusting (b) Audio signal after adjusting

Fig. 11. Audio signals before and after adjusting

(3) Adding noise attack: Use Cubase sequencer to write messy MIDI commands on different tracks of watermarked MIDI music. The MIDI music diagram after adding noise is shown in Fig. 12. Watermark extraction is performed after adding noise to the MIDI music, to evaluate the resistance against noise attack. The experimental results are shown in Table 2. After adding several other MIDI commands to the main melody track, the two MIDI music watermarking algorithms can still extract valid watermark images. Adding noise commands to the MIDI music sub-melody has no effect to the watermark extraction. We can extract all effective watermark images, which indicates that the proposed algorithms can resist noise attacks.

Fig. 12. Results after adding several random music information into the first track

Table 2. Extracted watermarks after adding noise attacks

Adding noise attack	Degree of addition	The first method	The second method
Add main melody track	Trace addition	54	54
	Add several	18	18
	Add a lot	0	0
Add sub-melody track	Trace addition	64	64
	Add several	64	64
	Add a lot	64	64

(4) Tampering attack: Tampering attacks does not change the size of the MIDI music files. We modify the time difference parameter and velocity value parameter of the MIDI command, and tamper the audio after the MIDI command as shown in Fig. 13 and Fig. 14. Watermark extraction is performed on the tampered MIDI music to detect the anti-attack ability. To improve the observability of the experiment, we embed a watermark image of 40×40 size into MIDI music. The experimental results are shown in Table 3. We can see from the table that the watermark image extracted from the modified MIDI music of 1% command is clearly visible. When modifying 5% of MIDI music command, some information of the extracted watermark image is lost. When the modification command exceeds 15%, the audio is damaged seriously, thus we cannot extract valid watermark information.

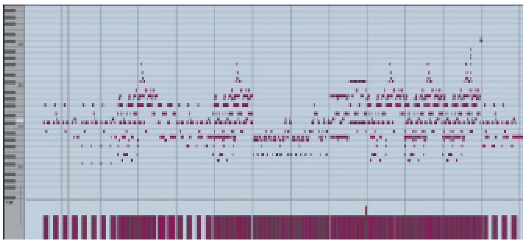


Fig. 13. MIDI music piano atlas

Fig. 14. Tamper the MIDI music

Table 3. Watermark results after tamper attacks

Modify the number of instructions	The first method	The second method
1%		
5%		
15%		

5.2 Subjective Quality Evaluation

The imperceptibility evaluation of the audio watermarking algorithm can be divided into two standards: objective quality evaluation and subjective quality evaluation. Objective quality evaluation is based on human visual and auditory characteristics, including average opinion score and subjective difference score. It is mainly used to measure the carrier distortion after embedding watermark [17-21].

The subjective evaluation method used in this paper is the Mean Opinion Score (MOS) as shown in Table 4 and Table 5. Five people were tested in this experiment. Watermarks with different capacities are embedded into the MIDI music. The MIDI music quality is almost unchanged. The human auditory is almost indistinguishable. If we embed the watermark with large capacity, it will change the MIDI music instruction parameters of the lowest three bits. Because the change range is large, the MIDI music quality will be seriously affected.

Table 4. MOS of the first method

Embedded image size	30×30	40×40	50×50
The first person	5	5	4
The second person	5	4	3
The third person	4	3	3
The fourth person	4	3	2
The fifth person	4	3	2

Table 5. MOS of the second method

Embedded image size	30×30	40×40	50×50
The first person	4	4	2
The second person	5	4	2
The third person	4	3	3
The fourth person	5	4	3
The fifth person	5	5	2

5.3 Objective Quality Evaluation

Because a MIDI music is not a sampled data, it is impossible to verify the transparency of the watermark by the Signal to Noise Ratio (SNR) or the Peak Signal to Noise Ratio (PSNR). To analyze the MIDI music data structure, we use the Mean Square Error (MSE) method and the normalized correlation coefficient method to verify the transparency of the MIDI music watermark. We propose an absolute entropy model to evaluate the quality of MIDI music.

(1) MSE method

The MSE is a more convenient method to measure the average error. It can evaluate the degree of the change in the data. When describing the experimental data, the less the MSE value is, the more accurate the prediction model is [22-25]

$$MSE = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(g(x,y)-g^{'}(x,y))^2}{M \times N}, \tag{22}$$

where the $g(x,y)$ is the original MIDI music, the $g^{'}(x,y)$ is the watermarked MIDI music, the M is the number of MIDI music data lines, and the N is the number of MIDI music data columns.

(2) Normalized correlation coefficient method

The normalized correlation coefficient is a dimensionless index. It is used to measure the linear closeness between two groups of variables in multivariate statistics, with a value range of [−1, 1]. It is divided into three types: positive correlation, irrelevance and negative correlation. Generally, the negative correlation and the positive correlation are combined together. The larger the value is, the stronger the correlation is. When the value is 1, the two vectors are exactly the same [26-27].

When the watermarked carrier audio is not attacked by other signals, the normalization coefficient of the extracted watermark image and the original image is 1. When the watermarked audio is subjected to other intentional or unintentional attacks during transmission, the normalized coefficient between the extracted watermark image and the original watermark image will decrease. The larger the normalization coefficient is, the higher the integrity of the watermark image remains. The MIDI music data structure is in the form of instructions, and the instruction range is between 0 and 127. In the correlation coefficient method, if the normalized correlation coefficient between the original MIDI music and the watermarked audio is closer to 1, the transparency of the MIDI watermark algorithm is better.

(3) Information entropy method

Information entropy indicates the average amount of information provided by each source symbol. Before output the source, it indicates the average uncertainty of the source. When the uncertainty is removed after the output of the source, entropy can be regarded as the amount of information that need to remove the uncertainty of the source [28-31].

Continuous source means that the messages sent by the source are formed by stochastic processes. For example, the voice signal $X(t,\omega)$ is not only continuous in amplitude, but also continuous in time. It belongs to an infinite set. A continuous variable can always be approximated by simplifying a discrete variable using digital quantization, and the smaller the quantization unit, the closer the resulting discrete variable is to that continuous variable. Therefore, the absolute entropy formula of continuous variables is defined as follows:

$$H_c(X) \overset{\text{def}}{=} -\int_R p(x)\, log\, p\,(x) dx, \tag{23}$$

If the $H(x)$ is large, randomness is large. The structure of the MIDI information command is composed of time difference, state, pitch and velocity. Thus, each MIDI pronunciation command can be regarded as discrete between commands and continuous within the command. With the absolute information entropy of continuous sources to determine the velocity value between the MIDI commands, the amount of change achieves an objective evaluation of the quality of MIDI music. The probability distribution diagram between MIDI commands is shown in Fig. 15. The absolute entropy calculation formula for MIDI music velocity values is defined as follows:

$$h(X) = -\int_{t1}^{t2} \frac{y}{t2-t1} log(\frac{y}{t2-t1}) dt, \tag{24}$$

$$\bar{h} = \frac{\sum_{i=1}^{N}(h_i(X))}{N}, \tag{25}$$

where the N is the number of information entropies calculated between instructions.

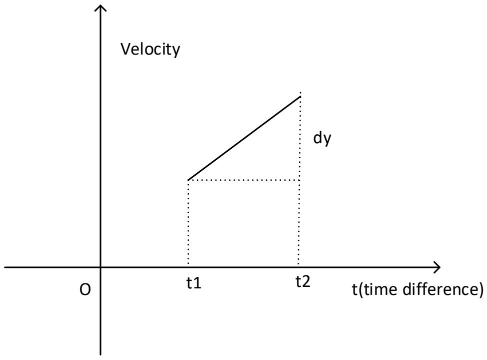


Fig. 15. The probability distribution diagram between MIDI commands

We can see from Table 6 and Table 7 that when MIDI music is embedded with 30 × 30 and 40 × 40 watermark, it has little effect on audio sound quality. Music professionals can hardly perceive the damage of the sound quality. When MSE<0.60 and correlation>0.99, the MIDI music quality is almost not damaged. The absolute average information entropy model data proposed in this paper has a good sound effect when it is greater than 48.57, the listeners hardly feel the noise. The two objective evaluation results are consistent with the subjective evaluation results, and the results obtained by the evaluation model proposed in this paper are consistent with the results of the subjective evaluation. As a result, the proposed algorithms have good transparencies.

Table 6. Objective quality rating table of the first method

Change in digits	Lowest 1	Lowest 2	Lowest 3
Image size	30×30	40×40	50×50
MSE	0.02	0.13	0.60
Entropy	55.91	48.61	41.98
Correlation	1.00	0.99	0.97

Table 7. Objective quality rating table of the second method

Change in digits	Lowest 1	Lowest 2	Lowest 3
Image size	30×30	40×40	50×50
MSE	0.02	0.13	0.61
Entropy	56.24	48.57	42.00
Correlation	1.00	0.99	0.97

6 Conclusion

In this paper, we propose two MIDI music copyright protection methods. The first one is a dual threshold audio watermarking algorithm based on MIDI note time difference and velocity value. Because of using the adaptive threshold in the embedding process, the transparency is significantly improved. With the Arnold scrambling operation, we further ensure the security of the watermark. The second one is a MIDI music watermarking algorithm based on DCT transform. Because of the DC components of DCT coefficients are selected to embed watermark, the performance of the algorithm remains high robustness. The watermark is embedded into the MIDI pitch, thus the we can achieve high concealment. Four kinds of attack experiments were carried out to verify the robustness of the proposed algorithms. The subjective and objective quality evaluations were carried out to verify the imperceptibility and the transparency. We also proposed a MIDI music quality evaluation scheme based on absolute information entropy. Experimental results show that the two MIDI music watermarking algorithms can resist sequencer attack to a certain extent. When the MIDI music is changed within the lowest 3 bit, the absolute information entropy is greater than 48.57, showing good transparency, and the human can hardly distinguish it. The proposed methods can extract effective watermark images after the cutting attacks, adjusting audio track order attacks, adding noise attacks and tampering attacks, showing good robustness. However, when the cutting ratio is about 75%, the watermark is difficult to be extracted. In the future work, we will further improve the performance of the MIDI music watermarking algorithms by analyzing feature signal of MIDI music with convolutional neural networks, and we will apply the proposed algorithms into practical environment to achieve copyright protection of MIDI music.

Acknowledgments

This research project was supported by the National Natural Science Foundation of China (Grant No. 62062064) and the Education Department of Jilin Province (Grant No. JJKH20200511KJ).

References

[1] D.M. Huber, The MIDI manual: a practical guide to MIDI in the project studio, third ed., CRC Press, New York, 2007.

[2] G. Hua, J.-W. Huang, Y.-Q. Shi, J. Goh, V.L.L. Thing, Twenty years of digital audio watermarking—a comprehensive review, Signal Processing 128(2016) 222-242.

[3] M. Hrncar, J. Krajcovic, Principles of Audio Watermarking, Advances in Electrical and Electronic Engineering (2011) 247-249.

[4] A. Tiwari, L. Jain, Digital Audio Watermarking: An Overview, Journal of Communication Engineering & Systems 5(1) (2015) 1-9.

[5] S. Srivastava, M. Chandra, G. Sahoo, Robust Voiceprint Based Audio Watermarking Using Wavelet Transform, in: Proc. Nanoelectronics,Circuits and Communication Systems, 2019.

[6] J.C. Hardwick, D.W. Griffin, Audio watermarking via phase modification: U.S. Patent No. 9990928, Jun 5, 2018.

[7] A. Kaur, M.K. Dutta, K.M. Soni, N. Taneja, Localized & self adaptive audio watermarking algorithm in the wavelet domain, Journal of Information Security and Applications 33(2017) 1-15.

[8] N. Cvejic, T. Seppanen, Increasing the capacity of LSB-based audio steganography, in: Proc. IEEE Workshop on Multimedia Signal Processing, 2002.

[9] A. Binny, M. Koilakuntla, Hiding secret information using LSB based audio steganography, in: Proc. 2014 Interna-tional Conference on Soft Computing and Machine Intelligence, 2014.

[10] C. Maha, E. Maher, K. Mohamed, B.A. Chokri, DCT based blind audio watermarking scheme, in: Proc. International conference on signal processing and multimedia applications (SIGMAP), 2010.

[11] Z. Liu, X. Zhao, Y. Jin, Audio Watermarking Algorithm for Tracing the Re-recorded Audio Source, IAENG International Journal of Computer Science 48(4)(2021) 1-8.

[12] G.-F. Chen, W.-J. Zhang, Watermarking Algorithm for Embedding MIDI Information into Digital Music Score, Journal of Shanghai University 14(2)(2008) 125-129.

[13] A. Adli, Z. Nakao, Three steganography algorithms for MIDI files, in: Proc. International Conference on Machine learning and cybernetics, 2005.

[14] D.-C. Wu, M.-Y. Chen, Information Hiding in Standard MIDI Files Based on Velocity Reference Values, International Journal of Network Security 18(2)(2016) 274-282.

[15] D.-C. Wu, M.-Y. Chen, Reversible data hiding in Standard MIDI Files by adjusting delta time values, Multimedia Tools and Applications 74(21)(2015) 9827-9844.

[16] K. Yamamoto, M. Iwakiri, A standard midi file steganography based on fluctuation of duration, in: Proc. International Conference on Availability, Reliability and Security, 2009.

[17] H. Fastl, Psychoacoustic basis of sound quality evaluation and sound engineering, in: Proc. 13th International Congress on Sound and Vibration ICSV13, 2006.

[18] R.C. Streijl, S. Winkler, D.S. Hands, Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives, Multimedia Systems 22(2)(2016) 213-227.

[19] C. Sloan, N. Harte, D. Kelly, A.C. Kokaram, A. Hines, Objective Assessment of Perceptual Audio Quality Using ViSQOLAudio, IEEE Transactions on Broadcasting 63(4)(2017) 693-705.

[20] T.-L.J. Yu, R.S. Schlauch, Diagnostic precision of open- versus closed-set word recognition, The Journal of the Acoustical Society of America 145(3)(2019) 1790.

[21] H. Axelsson, A. Bergstrom, P. Bjorken, P. Bruin, M. Sundberg, Improved Latency Performance with GSM/EDGE con-tinued evolution, in: Proc. IEEE Vehicular Technology Conference, 2016.

[22] S. L. Bagoussel, M. Paquier, C. Colomes, S. Moulin, Sound quality evaluation based on attributes - Application to binaural contents, 131st Audio Engineering Society Convention (2011) 8542.

[23] F. Rund, K. Ulovec, Comparison of two objective methods of quality assessment for digital audio broadcasting, in: Proc. 2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA), 2018.

[24] K. Ulovec, M. Smutny, Perceived audio quality analysis in digital audio broadcasting plus system based on PEAQ, Radioengineering 27(1)(2018) 342-352.

[25] J.-S. Mei, S.-K. Li, X.-M. Tan, A digital watermarking algorithm based on DCT and DWT, in: Proc. The 2009 International Symposium on Web Information Systems and Applications (WISA 2009), 2009.

[26] H.-X. Wang, K. Ding, C.-X. Liao, Chaotic watermarking scheme for authentication of JPEG Images, in: proc. 2008 International Symposium on Biometrics and Security Technologies, 2008.

[27] W. Al-Nuaimy, M.A.M. El-Bendary, A. Shafik, F. Shawki, A.E. Abou-El-azm, N.A. El-Fishawy, S.M. Elhalafawy, S.M. Diab, B.M. Sallam, F.E. Abd El-Samie, H.B. Kazemian, An SVD audio watermarking approach using chaotic encrypted images, Digital Signal Processing 21(6)(2011) 764-779.

[28] Q.-C. Zhong, Q.-X. Zhu, A DCT domain color watermarking scheme based on chaos and multilayer Arnold transformation, in: Proc. 2009 International Conference on Networking and Digital Society, 2009.

[29] O. Jane, H.G. İlk, E. Elbaşı, A secure and robust watermarking algorithm based on the combination of DWT, SVD, and LU decomposition with Arnold’s Cat Map approach, in: Proc. 2013 8th International Conference on Electrical and Electronics Engineering (ELECO), 2013.

[30] J.-C. Zhou, X.-J. Tie, Arnold Transformation of Digital Image with Two Dimensions and its Periodicity, Journal of North China University of Technology 12(1)(2000) 10-14.

[31] D. Inoue, T. Matsumoto, Scheme of standard MIDI files steganography and its evaluation, in: Proc. Security and Watermarking of Multimedia Contents IV, 2002.