Audio Steganography and Watermark

Zhang Jingmiao
University of Science and Technology of China
nanshan@mail.ustc.edu.cn

Wei Panyue
Huazhong University of Science and Technology
weipp7@gmail.com

Wang Junchao
Sichuan University
junchaowang613@gmail.com

Wei Xinyue
Xi'an Jiaotong University
wxyeipai@stu.xjtu.edu.cn

ABSTRACT
This paper proposes methods to apply LSB, DWT, DWT-LSB and DCT algorithms to audio digital watermarking and explores their robustness respectively. Among them, the LSB method is based on the replacement of the least significant bits, the DWT is based on the Discrete Wavelet Transform, the DWT-LSB is the combination of DWT and LSB, and the DCT is based on the Discrete Cosine Transform. To evaluate their robustness, we tested their performance for several audio manipulations, and overall they sometimes perform well and sometimes perform poorly. We also propose the whole process of adding, extracting, and verifying digital watermarks in practical application scenarios, and LSB watermarking can be competent in this scenario.

Keywords
Watermarking, Cryptography, Copyright Protection, Robustness

1. INTRODUCTION
Digital watermarking technology is an information hiding technology. The so-called audio digital watermarking algorithm is to embed a digital watermark into an audio file (such as .wav, .mp3, .avi, etc.) through a watermark embedding algorithm, but it has no effect on the original sound quality of the audio file or the human ear can't feel its effect. On the contrary, through the watermark extraction algorithm, the audio digital watermark is completely extracted from the audio host file. The watermark must be robust to attacks and other types of distortion to prevent tampering and forgery. Typical attacks include adding noise, data compression, filtering, resampling, A/D-D/A conversion, statistical attacks, etc.

The digital watermark can be identified and recognized by the producer. Through the information hidden in the carrier, it can achieve the purpose of confirming the content creator, buyer, transmitting secret information or judging whether the carrier has been tampered with. Digital watermarking is an effective way to protect the information security, and realize anti-counterfeiting traceability, and copyright protection. It is an important branch and research direction in the field of information hiding technology.

We apply LSB, DWT, DWT-LSB and DCT methods to audio watermarking, respectively, and discuss their robustness and applicable scenarios. Since they are not guaranteed to be completely indestructible and tampered with, we also propose an application method in a double-ended scenario to ensure audio integrity and copyright information.

2. PREVIOUS WORKS
There is a rich body of literature on audio digital watermarking technology. As early as 1996, Laurence Boney et al proposed a digital watermark for audio signals [1]. Ingemar J. Cox et al proposed a method for inserting watermarks in the frequency domain [2]. The combination of the DCT and LSB method is regarded as a fragile watermark [3]. Lee, S. J. and Jung, S. H. introduced the wide application of DWT in audio watermarking [4].

3. OUTLINE
We introduce the LSB, DWT, DWT-LSB and DCT algorithms in §4 and propose methods to evaluate their robustness. In §5 we introduce the practical application process and robustness analysis of the four algorithms. §6 proposes a secure watermarking and verification scenario. Finally, §7 summarizes the above work.

4. AUDIO WATERMARKING ALGORITHMS

4.1 Least Significant Bits
The basic idea of LSB replacement steganography is to replace the lowest bit of the carrier image with the secret information to be embedded. When the information is embedded, the probability of the lowest bit being changed is at most 50%. Even so, the LSB replacement steganography only introduces very little noise in the original image, which is inaudible. The audio file is encoded into an 8-column binary matrix, and changing its lowest bit has little effect on the sound quality. Therefore, embedding the binary code of the watermark file in these positions can achieve the effect of steganography.

LSB replacement steganography can be described as follows: First, the following notation is introduced: c represents the original audio (carrier) in the embedding process, which can be represented by a sequence c_i of length $l(c)$; s represents the audio after embedding the watermark, which can also be regarded as a sequence s_i of length $l(c)$, $1 \leq i \leq l(c)$; m represents the secret message to be embedded, which is a sequence of m_i with a length of $l(m)$, $l(m) \leq l(c)$, generally speaking, $m_i \in \{0, 1\}$. j represents the index value of audio. j_i indicates the order of index values. c_{ji} represents the j_i carrier element. k represents the steganography key.

The embedding and extracting process of LSB replacement steganography is as Algorithm 1:

Algorithm 1 LSB Watermarking Algorithm

1: Select the first $l(m)$ replaceable bits in the audio.
2: each bit selected, if its LSB is the same as the information bit to be embedded, do not change it; otherwise, proceed to the next step;
3: Replace the LSB of the original positions with the watermark information bits, while the high 7 bits remain unchanged, and the modified image is s.
4: Extract the LSB of the first $l(m)$ audio bits and arrange them to form secret information m.

4.2 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT), receives a discrete signal as input $x(n)$ and outputs another discrete signal. The general idea of DWT is to separate the high-frequency and low-frequency signals with two filters, the higher filter and the lower filter. After the separation process, the array $x[n]$ which has a length of N, is transformed into two arrays, x_H and x_L, whose lengths are both $N/2$. X_H represents the high-frequency signals, and x_L refers to the low-frequency signals.

There are two functions $h(x)$ and $g(x)$ used as filters. From $x[n], n = 1, 2, \ldots, N$, we calculate the low-frequency signal and the high-frequency signal in the following way:

$$x_{1,L}[n] = \sum_{k=0}^{K-1} x[2n-k]g[k] \tag{1}$$

$$x_{1,H}[n] = \sum_{k=0}^{K-1} x[2n-k]h[k] \tag{2}$$

This process can continue to extend, as shown in Figure 1.

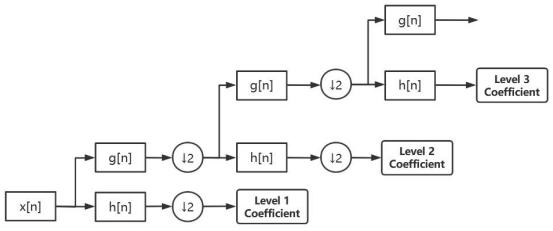


Figure 1: Extension of DWT Signal

In theory, the process can extend indefinitely. While in practice, the number of filtered layers can be determined depending on the specific case. In our project, the number of filtered layers is 2, for if we use more layers, the fewer information can be hidden, but if we use fewer layers it can store more information but may influence the primary audio. Notably, this process is theoretically fully recoverable, so when we get the array after DWT, we can recover the primary array.

Audio can be converted to an integer array, which is the 1-dimension discrete signal. We use a short integer array, in which every value is between 0 and 256, to present a grey image. After we use 2-level-DWT to the audio array, we get high-frequency signals in the second level. If we replace this array partly with the short integer array which stands for the picture and then use iDWT to recover the array and then the audio, it sounds quite the same as the primary audio. That is to say, we successfully hide an image into audio, for the listener can't notice that.

The process of adding and extracting the watermark of the DWT algorithm is shown in Algorithm 2. In the third step, notice that the length of $cD2$ and the image array are not always the same, we just replace a few elements in the front. For example, if $cD2$ = [1,2,3], image array = [4,5,6,7,8], then after the replacement, $cD2$ = [1,2,3,7,8].

Algorithm 2 DWT Watermarking Algorithm

1: Read the audio file and get an array of integer.
2: Use DWT to transform this int array to 3 arrays named $cA2$, $cD2$ and $cD1$.
3: Transform the watermark image as an integer image array, and replace $cD2$ with the array.
4: Use iDWT to transform the three arrays($cA2$, $cD2$, $cD1$) into one array.
5: Transform the array to audio.
6: The extraction process is its inverse transformation.

4.3 DWT-LSB

Even though the DWT process is recoverable theoretical, in actual calculation the primary audio and the recovered audio can't be all the same. That is because in the process of calculating the answer can't always be an integer. And when the computer deals with floats, some bad things happen! However, it's quite difficult for the human eyes to find the tiny difference between the 123-degree grey and 124-degree grey, so it's not a serious problem. But if we use the same way to hide a string into the audio, then the tiny difference can cause a false decryption result. In comparison, LSB can perfectly recover the hidden watermark, but it has worse robustness. Our group comes up with a new method to hide a watermark in audio which combines the merits of DWT and LSB. We named it the "DWT-LSB" method. The idea can be summarized as Algorithm 3.

4.4 Discrete Cosine Transform

Discrete Cosine Transform (DCT) is a transformation related to Fourier, similar to discrete Fourier transformation, but only uses real numbers. The idea is to convert the audio from the time domain to the frequency domain via the DCT, look for the chunks of data in the frequency domain that cause the smallest overall change, and overwrite it with our watermark. In this study, we tried to block the audio and then DCT transform all the audio blocks. In the DCT

Algorithm 3 DWT-LSB Watermarking Algorithm

1: Convert the image into an array of bits $img[i]$, and convert the audio into an int array $audio[i]$.
2: Use DWT to the audio int array and change the array in the following way:
3: For i in range $(0, len(img_bit_array))$:
4: If the $image\_array$ of bits $img[i] = 0$, then change $audio[i \times tms]$ into a small number smn;
5: If the $image\_array$ of bits $img[i] = 1$, then change $audio[i \times tms]$ into a big number, bgn. smn, bgn and tms are three self-selectable constants: tms: the bigger the better, if the $audio\_array$ length is enough. In our project we set $smn = 10$, $bgn = 100$, $tms = 2$.
6: Using iDWT to recover the $audio\_array$ and change it to audio.

domain, watermark components of different intensities are embedded in the DCT coefficients of image blocks according to the result of block classification. According to the characteristics of the human sensory system, we choose to embed the watermark signal into the intermediate frequency coefficients of the original audio to retain better robustness and invisibility.

The schematic diagram of DCT transformation is shown in Figure 2.

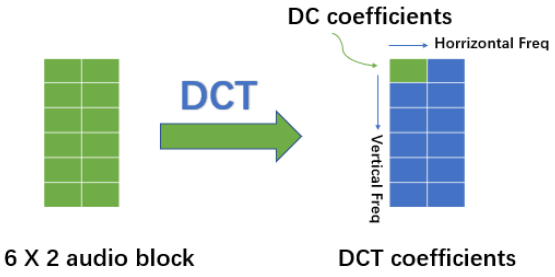


Figure 2: DCT Transformation

Formulae for DCT and inverse DCT:

$$F(u,v) = \frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos[\frac{(2x+1)u\pi}{2N}]cos[\frac{(2y+1)v\pi}{2N}] \tag{3}$$

$$f(i,j) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)F(u,v)cos[\frac{(2x+1)u\pi}{2N}]cos[\frac{(2y+1)v\pi}{2N}] \tag{4}$$

The overall process of DCT algorithm is shown in Algorithm 4.

4.5 Robustness Analysis

The LSB algorithm embeds the watermark information into the least significant bit of the data, that is, the LSB of the data is replaced by the watermark information that needs to be added. This is precisely because the low-level data has the least impact on the overall data, which also leads to the low strength of the embedded watermark information, otherwise, it will affect the data quality of the carrier.

Algorithm 4 DCT Watermarking Algorithm

1: Divide the audio into non-overlapping blocks of size $row \times collum$.
2: Suppose we have n such blocks.
3: Generate a set of keys called k. This will be used as a seed for the pseudo-random generator.
4: The size of the watermark is expressed as width w times height h. Therefore, there are $fea = w \times h$ feature points in the watermark.
5: Check if $n \leq fea$. If not, the watermark cannot be embedded.
6: For each feature point of the watermark, repeat 7-10:
7: Compute the DCT of this block. Let it be named dct_block.
8: Let $change[i] = k[i] \times watermark[h][w]$.
9: Add the $change[i]$ to the last column of the current block, $dct\_block[i, collum - 1] += change[i]$.
10: Calculate the inverse DCT of the block and save it.
11: Save watermarked audio.

Therefore, the algorithm is only used for fragile digital watermarking (compared with robust digital watermarking, it can not bear a lot of distortion).

By using DWT we can separate the high-frequency signals and the low-frequency signals, or the important signals and the less important signals. If we change the content in $cD1$ and $cD2$, the details of the DWT coefficients of the primary audio can't change much. When we do some bad things to the primary audio, such as add noise into it, the detail coefficients might change a little, but we replace whole integers instead of using just one bit, so when the integers change a little, the image can be shown too. Even though they can't be the same as the primary watermark, they look quite similar.

For the method combining DWT and LSB, when extracting the watermark from the encrypted audio, we compare $audio\_array[i]$ with $\frac{smn+bgn}{2}$ to decide what the current bit is. Although the calculation deviation remains, the absolute deviatation is usually smaller than 10, so the result of comparing $audio\_array[i]$ with $\frac{smn+bgn}{2}$ can hardly be changed. In this way, the watermark recovered and the primary watermark can be the same, and the audio has better robustness.

We perform DCT on the audio and obtain a DCT coefficient matrix, then embed the image information into the matrix in blocks. We chose the block positions that had the least impact on the whole data for embedding. Thus, when we attack our audio with a watermark, such as add noise or shrink it, we compute $final = IDCT(DCT)$ and right now the final contains the original floating-point values and some little noise. But when we save it as audio, the floating-point numbers are converted into integers. When the noise is small, the final integer in the audio doesn't change. But it all depends on the size of the noise, if the noise is big, it will change the integers in the audio.

In §5.3, we will evaluate the robustness of the four watermarking algorithms through actual attack cases. The final result will show the attack strength and robustness.

5. IMPLEMENTATION

5.1 Adding Watermarks

We use text as LSB watermark, image as DWT watermark, and image and audio as DCT watermark to seek diversity.

Using the LSB algorithm, we encode the audio into 0-1 matrix and replace the lowest bits to get the audio file with a watermark, as shown in the Figure 3.

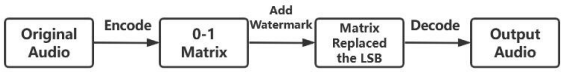


Figure 3: LSB Watermark Addition

We perform DCT/DWT on the audio to obtain a DCT/DWT coefficient matrix, embed the image information into the matrix in blocks, and then perform IDCT/IDWT on it to obtain audio with a watermark, as shown in the Figure 4.

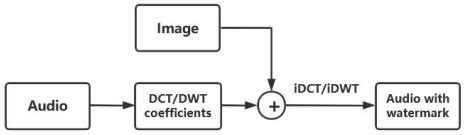


Figure 4: DCT/DWT Watermark Addition

5.2 Extracting Watermarks

The extraction process of the LSB watermark is opposite to that of adding, and the lowest bit needs to be replaced, as shown in the Figure 5.

Figure 5: LSB Watermark Extraction

We perform DCT/DWT on the audio with a watermark to obtain a DCT/DWT coefficient matrix, then we decrypt the block of matrix and merge these pieces of information. After that, we got the extracted watermark, as shown in the Figure 6.

5.3 Attack & Robustness Analysis

We demonstrate the attacks of noise adding, cropping, merging, reverberation, filtering and compression, and put some samples here. If the text can be 90% restored, the image NC coefficient is greater than 0.5, and the audio correlation coefficient is greater than 0.9, we believe that it is robust under specific attacks, and vice versa. If the watermark is robust, we indicate "yes" with a "Y" in the "robustness" row. If not, we'll note the "N" in the "Robustness" line for "no".

1. **Noise Adding**
   Taking DWT as an example, the DWT original watermark is shown in Figure 7, and the DWT extracted

Figure 6: DCT/DWT Watermark Extraction

watermark without attack is shown in Figure 8. Gaussian noise with a signal-to-noise ratio (SNR) of 10, 50, 100, or 150 is added to the embedded audio, and then the watermark is extracted. The results are shown in the Figure 9.

Figure 7: DWT Original Watermark

Figure 8: DWT Watermark Extraction Without Attack

The NC coefficient of these extracted watermarks compared to the original image is 0.9906 when there is no attack, and 0.0206, 0.8554, 0.9906, and 0.9906 when the SNR is 10, 50, 100, and 150, respectively.

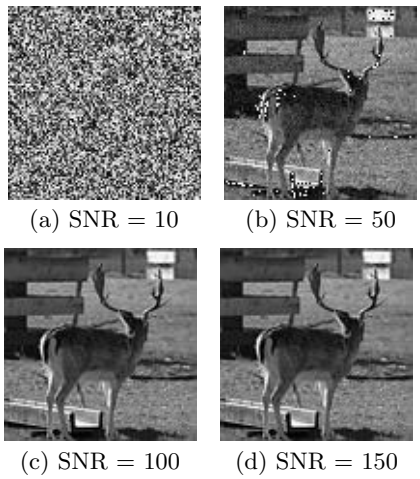
However, the robustness of the LSB text watermark and DCT audio watermark after adding noise is very poor, the LSB text is all garbled, and the DCT audio correlation coefficient is less than 0.1. All results are shown in the Table 1.

As we expected, the robustness of the DWT-LSB algorithm is better than that of the DWT algorithm when the SNR is large. Figure 10 shows the NC coefficients of the two methods at different SNRs.

2. **Cropping**
   Taking LSB as an example, the LSB watermark is shown in Figure 11, which can be 100% restored without attack. We crop the rear of the audio, and the text watermark restoration rate can reach more than 95%, as shown in the Figure 12. However, if the front part of the audio is cropped, garbled characters are extracted.

   For DWT, DCT and DWT-LSB, if the audio end segment is cut off, the watermark image is either not affected or cannot be generated (the array length is not enough and the program reports an error); if the audio header segment is cut, the watermark image cannot be generated. All results are shown in the Table 2.

3. **Merging**
   Merging and cropping are similar. The experimental



(a) SNR = 10 (b) SNR = 50

(c) SNR = 100 (d) SNR = 150

Figure 9: Watermark Extracted After Noise

Table 1: Performance Against Adding Noise

SNR	10	50	100	150
LSB	N	N	N	N
DWT	N	Y	Y	Y
DCT	N	N	N	N
DWT-LSB	N	Y	Y	Y

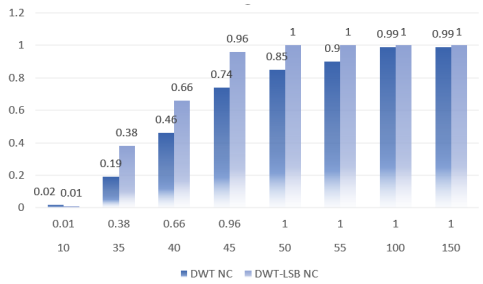
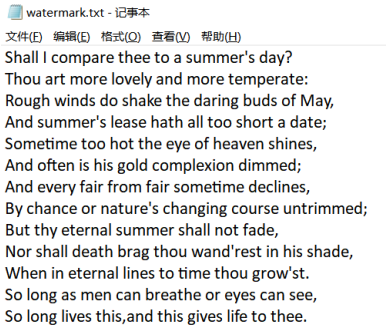


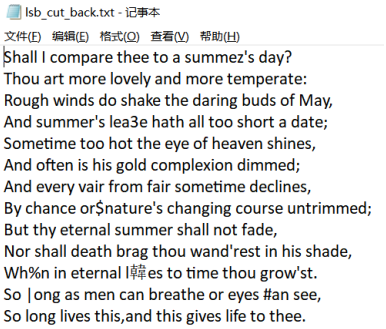
Figure 10: Comparison of DWT-LSB and DWT NC Coefficients


watermark.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the daring buds of May,
And summer's lease hath all too short a date;
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimmed;
And every fair from fair sometime declines,
By chance or nature's changing course untrimmed;
But thy eternal summer shall not fade,
Nor shall death brag thou wand'rest in his shade,
When in eternal lines to time thou grow'st.
So long as men can breathe or eyes can see,
So long lives this,and this gives life to thee.

Figure 11: LSB Original Watermark


lsb_cut_back.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a summez's day?
Thou art more lovely and more temperate:
Rough winds do shake the daring buds of May,
And summer's lea3e hath all too short a date;
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimmed;
And every vair from fair sometime declines,
By chance or$nature's changing course untrimmed;
But thy eternal summer shall not fade,
Nor shall death brag thou wand'rest in his shade,
Wh%n in eternal l韓es to time thou grow'st.
So |ong as men can breathe or eyes #an see,
So long lives this,and this gives life to thee.

Figure 12: LSB Watermark After Tailoring

Table 2: Performance Against Cropping

Cropping	Rear	Front
LSB	Y	N
DWT	Y	N
DCT	Y	N
DWT-LSB	Y	N

results show that when a piece of audio is added to the back of the carrier audio, the robustness is good, but when a piece of audio is added to the front, the robustness is bad.

For DCT audio watermarking, the correlation coefficient before and after the attack is 0.9999. The waveforms are shown in Figure 13 and Figure 14. All results are shown in the Table 3.

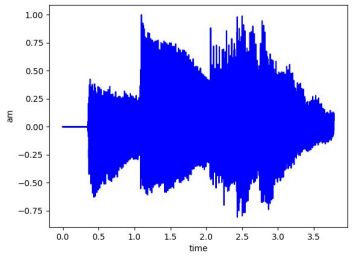


Figure 13: DCT Original Watermark

4. **Volume Changing**
Changing the volume also affects watermark extraction, and we find that DWT and DWT-LSB algorithms perform better in this attack. Figure 16 is the performance of the DWT watermark when the volume is increased or decreased by 2, 4 or 6 dB, and their NC coefficients are 0.7831, 0.6595, 0.1134, 0.9908, 0.9834, and 0.9772, respectively. Figure 17 is the performance of DWT-LSB under the same attack, and their NC coefficients are 0.9987, 0.9706, 0.8854, 1.0, 1.0, and 0.0674, respectively. DWT-LSB does not perform as well as DWT in the direction of audio volume reduction when the reduction range is large; it performs better than DWT in other audio volume adjustment directions. For space reasons, the pictures are in the appendix.

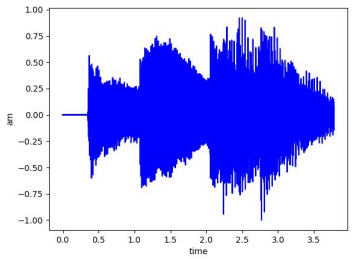


Figure 14: DCT Watermark After Merging

Table 3: Performance Against Merging

Merging	Rear	Front
LSB	Y	N
DWT	Y	N
DCT	Y	N
DWT-LSB	Y	N

LSB and DCT perform poorly on volume attacks. The overall results are shown in Table 4.

5. **Other Attacks**
We also tried filtering, reverberation, and compression attacks, all of which are not robust, and the results are shown in the Table 5.

6. A DOUBLE ENDED SCENARIO

As mentioned in the previous section, none of the four watermarking algorithms can guarantee 100% robustness, and copyright information will be destroyed under severe attacks. Therefore, we propose a watermarking and verification method in a two-terminal communication scenario, which can ensure the security of copyright information. The communication process is shown in the Figure 15. The black line indicates Alice's action and the red line indicates Bob's action.

Alice uses the SHA256 algorithm to obtain the message digest of her copyright information, and then RSA signature and AES encryption to get the final watermark, which is embedded in the audio and sent to Bob, and the initial copyright information is also sent to Bob.

After receiving the message, bob extracts the watermark from the audio containing the watermark, decrypts it, designatures it, and obtains the message digest. On the other hand, he compares the SHA256 digest of the original copyright information with the result obtained in the previous step. If it is the same, the verification passes, otherwise, it

Table 4: Performance Against Changing Volume

Changing Volume(dB)	+2	+4	+6	-2	-4	-6
LSB	N	N	N	N	N	N
DWT	Y	Y	N	Y	Y	Y
DWT-LSB	Y	Y	Y	Y	Y	N
DCT	N	N	N	N	N	N

Table 5: Performance Against Other Attacks

	Filtering	Reverberation	Compression
LSB	N	N	N
DWT	N	N	N
DCT	N	N	N
DWT-LSB	N	N	N

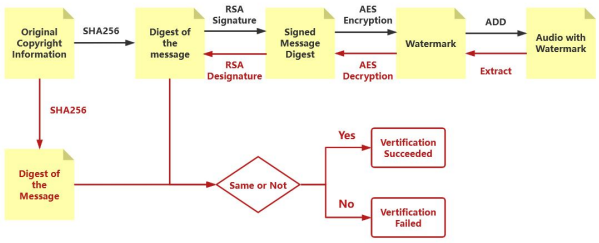


Figure 15: A Double Ended Scenario

fails.

The simultaneous use of hashing, signing and encryption ensures the data integrity and confidentiality of this protocol. In this way, the copyright information is verifiable and unforgeable.

7. CONCLUSIONS

First, we implemented three audio watermark embedding algorithms, LSB, DWT and DCT. Then, to ensure the security of watermarking, we applied them to two-terminal communication scenarios and proposed a DWT-LSB algorithm optimized based on DWT. Finally, we compare the robustness of LSB, DWT, DCT and DWT-LSB. The better robustness of DWT-LSB justifies our improved new method.

8. ACKNOWLEDGMENTS

Thank Prof. Hugh and TA Harish for their help in our project, and thank all the team members for their cooperation.

9. REFERENCES

[1] L. Boney, A. Tewfik, and K. Hamdy. Digital watermarks for audio signals. In *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pages 473–480, 1996.

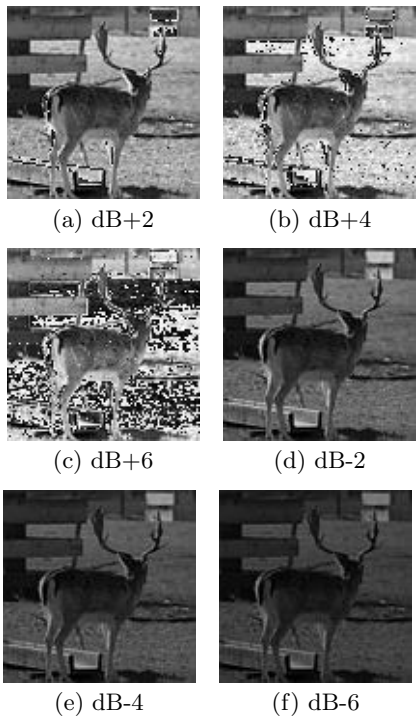
[2] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. A secure, imperceptible yet perceptually salient, spread spectrum watermark for multimedia. In *Southcon/96 Conference Record*, pages 192–197, 1996.

[3] J. Fridrich and M. Goljan. Images with self-correcting capabilities. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 792–796 vol.3, 1999.

[4] S. J. Lee, S. H. Jung, and IEEE. A survey of watermarking techniques applied to multimedia, 2001.

APPENDIX

A. WATERMARK UNDER CHANGING VOLUME ATTACK

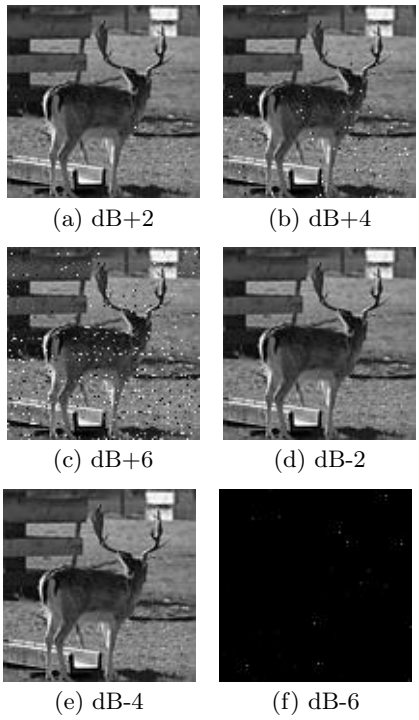


(a) dB+2 (b) dB+4

(c) dB+6 (d) dB-2

(e) dB-4 (f) dB-6

Figure 16: DWT Watermark Under Changing Volume Attack



(a) dB+2 (b) dB+4

(c) dB+6 (d) dB-2

(e) dB-4 (f) dB-6

Figure 17: DWT-LSB Watermark Under Changing Volume Attack