Detecting MP3Stego using Calibrated Side Information Features

Xianmin Yu
School of Information Science and Engineering, Ningbo University
Email: mlhappy1016@163.com

Rangding Wang and Diqun Yan
School of Information Science and Engineering, Ningbo University
Email: {wangrangding@nbu.edu.cn, yandiqun@nbu.edu.cn}

***Abstract*—MP3Stego is a typical steganographic tool for MP3 audio, which embeds secret message into MP3 audio according to the parity of the block length. In this work, we present a steganalytic method to attack MP3Stego. The big_values inside information is considered to extract the steganalytic feature. Re-compression calibration has been applied in order to improve the feature's sensitivity. Experimental results show that the extracted feature can reflect the MP3Stego trace effectively.**

***Index Terms*—steganalysis, MP3, big_values, calibration**

I. Introduction

Steganography is the art of hiding the present of communication by embedding secret messages into innocent looking covers, such as digital images, videos, and audios [1]. Its aim is to avoid drawing suspicion to the transmission of hidden information. Steganalysis is the art of detecting secret messages hidden using steganography [2]. The goal of steganalysis is to reveal the presence of embedded message and to break the covert communication.

MP3, as a standard for transmission and storage of compressed audio, is a promising carrier format for steganography. First, MP3 is the most popular and widely used audio file format. When audios in MP3 format are taken as cover signals, the stego-audios will be less likely to be noticed by steganalyzers than other audio formats. Also, it is a challenge for steganalyzers to distinguish whether the distortion is caused by stego operation or by MP3 encoding since MP3 is a lossy compression algorithm. For these reasons, MP3 audio is easily to be chosen as carrier for steganography and hence the competition between MP3 steganography and MP3 steganalysis has escalated over the past few years. Several stego tools for MP3 audios have been arisen, such as MP3Stego [3], UnderMP3Cover [4], MP3Stegz [5], and Stego-Lame [6]. In the early works of our group members, we also have proposed two novel steganographic methods for MP3 audios [7, 8]. As far as MP3 steganography tools are concerned, MP3Stego is the most typical one. In MP3Stego, with the benefit of the distortion adjustment mechanism of the MP3 codec, the sum of the distortion caused by quantization and MP3Stego is effectively controlled below the masking threshold which is the minimum sound level that human can perceive. Once the cover audio is unavailable, it is hard to distinguish whether the test audio has been operated by MP3Stego or not.

In order to defeat MP3Stego, some steganalytic methods have been proposed in recent years. Since MP3Stego changes the behavior of quantization in MP3 encoding, Westfeld [9, 10] proposed an attack method based on the variance of the block length which is a parameter related to quantization. Similarly, Dittmann [11, 12] considered that there would be more different block lengths after MP3Stego embedding. Hence, the number of different block lengths is taken as a steganalytic feature in their method. However, the performance of this method can be still improved especially at low embedding rates. Qiao [13] introduced a detection method based on an inter-frame feature set which contains the moment statistical features on the second derivatives, as well as Markov transition features and neighboring joint density of MDCT coefficients. In his another work [14], the statistical moments of GGD (Generalized Gaussian Distribution) shape parameters for MDCT coefficients are also taken as the steganalytic features. Ozer et al. [15, 16] presented a method for detecting audio steganography based on audio quality metrics. Although this method can expose the presence of MP3Stego, relatively high false positive rate is one of the limitations. Additionally, the dimensionality of the feature space is the main inconvenience of Qiao’s and Ozer’s methods.

Generally, the steganalytic detector will be more straightforward and effective if the features can be extracted directly from the position where the steganography takes place. Since MP3Stego happens during quantization, more attentions should be paid to the parameters associated with the quantization. In this work, the big_values which represents the number of quantized MDCT (QMDCT) coefficients in Big_value region is introduced into steganalysis to attack MP3Stego. In order to weaken the influence of the audio content and expose the stego noise, the calibrated feature is extracted by recompression. The results demonstrate that the proposed

doi:10.4304/jsw.8.10.2628-2636

method can achieve a good discriminatory ability for MP3Stego.

The rest of this paper is organized as follows. Section II briefly covers the basic operations of the MP3Stego algorithm. Section III presents the proposed method. The effectiveness of the proposed method under various conditions with experimental results is verified in Section IV. Finally, conclusions are drawn in Section V.

II. BACKGROUND

In this section, the principle of MP3 compression is first reviewed in order to better understand MP3Stego and then the MP3Stego steganography algorithm are analyzed.

A. Principle of MP3 Compression

Pulse Code Modulation (PCM) is a standard format for storing or transmitting uncompressed digital audio. There are two parameters for PCM: sample rate Hz and bit-rate Bit. The sample rate describes how many samples per second, and the bit-rate describes how big the digital word is that will hold the sample value. MPEG-1 Audio Layer 3, or MP3, is an audio-specific format that was designed by the Moving Picture Experts Group as part of its MPEG-1 standard [17]. The lossy compression algorithm of MP3 is designed to greatly reduce the amount of data required to represent the audio recording with a faithful reproduction of the original uncompressed audio for most listeners. Fig.1 illustrates the entire MP3 encoding, which consists of the following steps [18]:

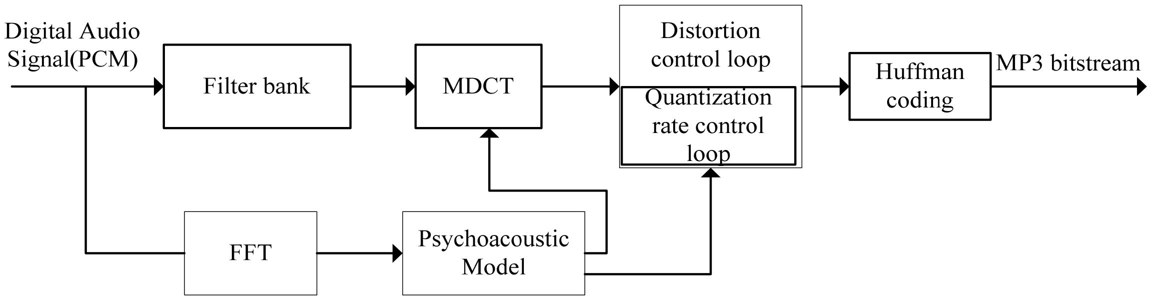


Figure 1. Block diagram of MP3 encoding

(1) Through a polyphase filter analysis, a sequence of 1152 PCM samples are filtered into 32 equally spaced frequency sub-bands depending of the Nyquist frequency of the PCM signal. If the sample frequency of the PCM signal is 44.1 kHz, the Nyquist frequency will be 22.05 kHz. Each sub-band will be approximately 22050/32 = 689Hz wide. The lowest sub-band will have a range from 0 to 689Hz, the next sub-band 689-1378Hz, etc.

(2) By applying a modified discrete cosine transform (MDCT) to each time frame of sub-band samples, the 32 sub-bands will be split into 18 finer sub-bands creating a granule with a total of 576 frequency lines. To reduce artifacts caused by the edges of the time-limited signal segment, each sub-band signal has to be windowed prior to the MDCT. If the sub-band signal at the present time frame shows little difference from the previous time frame, then the long window type is applied, which enhance the spectral resolution given by the MDCT. Alternatively, if the sub-band signal shows considerable difference from the previous time frame, then the short window is applied, which enhance the temporal resolution.

(3) Simultaneously, the signal is also transformed to the frequency domain by a Fast Fourier Transform as the signal is processed by the polyphase filterbank, to obtain higher frequency resolution and information on the spectral changes over time.

(4) Psychoacoustic model retrieves the frequency information from the FFT output and provides information about which parts of the audio signals that is audible and which parts are not. The two presently FFT spectra and the two previous spectra are compared. If considerable differences are found, a request of adopting short windows will be sent to the MDCT block. As soon as the difference fades away, the MDCT block will be changed back to long windows. At the same time, the Psychoacoustic model detects the dominant tonal components and for each critical band masking thresholds are calculated. Frequency components below the thresholds are masked out.

(5) The scaling and quantization are applied to 576 spectral values at a time, which is done iteratively in two nested loops, a distortion control loop(outer loop, which aims to keep the quantization noise below the masking threshold) and a rate control loop (inner loop, which determines the quantization step size and the quantization of the frequency domain samples).

(6) The Huffman coding, which retains MPEG-1 Layer III a high quality at low bit-rates, is applied to the quantized values. All parameters generated by the encoder reside in the side information part of the frame. The frame header, side information, CRC, Huffman coded frequency lines etc. are put together to form frames.

In filter bank analysis, a polyphase filter $H_i[n]$ is given by:

$$H_i[n] = h[n] \times \cos[\frac{\pi \times (2 \times i + 1) \times (n - 16)}{64}], i = 0 - 31, n = 0 - 511$$

(1)

Where $h[n]$ is a low-pass filter. The original PCM signal and filtered signals and filtered signals are denoted by $x[n]$ and $P_i[n]$ respectively.

$$P_i[n] = \sum_{m=0}^{511} x[n-m] \times H_i[m], i = 0 - 31$$

(2)

We obtain sub-band signals $S_i[n]$, which is $P_i[n]$ down-sampled by 32.

$$S_i[n] = P_i[32n] = \sum_{i=1}^{n} x[32n - m] \times H_i[m], i = 0 - 31$$

(3)

As a lapped transform, the MDCT is a bit unusual compared to other Fourier-related transforms in that it has half as many outputs as inputs (instead of the same number). To obtain MDCT coefficients, MP3 defines long window and short window to enhance frequency resolution and temporal resolution, respectively. The product of selected window coefficients and sub-band signal is denoted by Z_k . The MDCT coefficients are given by :

$$X_i = \sum_{k=0}^{n-1} Z_k \cos[\frac{\pi}{2n}(zk + 1 + \frac{n}{2}) \times (2i+1)], i = 0,1,\ldots,\frac{n}{2} - 1$$

(4)

For the long window, n equals 36, otherwise n equals 12 for the short window.

In Psychoacoustic Model II, $MNR(m)$ indicates the mask-to-noise ratio at m bits quantization, which measures the threshold of perceptible distortion. With the signal-to-noise ratio (SNR) of the signal, $SMR(m)$ is calculated as:

$$SMR(m) = SNR(m) - MNR(m) \qquad (5)$$

The MP3 compression algorithm relies on exploiting weakness of human auditory perception and hides the fact that a significant amount of information is discarded without any noticeable degradation of quality.

In MP3 compression, several bit-rates are specified in the MPEG-1 Layer 3 standard: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, and 320kbit/s, and the available sampling frequencies are 32, 44.1, and 48 kHz.

The MP3 decoding, demonstrated in Fig.2, is the inverse process of encoding contains the steps in the order of Huffman decoding, inverse quantization, inverse MDCT and aliasing cancellation, and filter-bank synthesis. First Huffman decoding is performed on the MP3 bitstream, then the decoder restore the quantized MDCT coefficient values and the side information related to them, such as the window type that is assigned to each frame. After inverse quantization, the coefficients are transformed back to the sub-band domain by applying an inverse MDCT on the coefficients. Finally, the waveform in the PCM format is reconstructed by the synthesis filterbank.

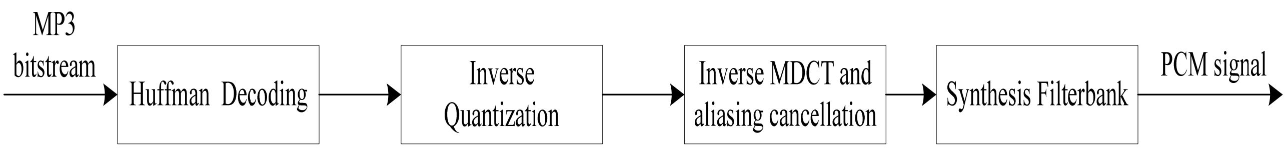


Figure 2. Block diagram of MP3 decoding

B. Review of MP3Stego Steganography Algorithm

The embedding of MP3Stego is integrated with the inner loop function that controls the bit rate of MP3 encoder. The pseudo-code for the inner loop function with MP3Stego is given in Fig. 3.

Algorithm: MP3Stego Embedding (inner loop function)

Input: MDCT Coefficients (I), Secret bit (b)

Input Parameters: Number of available bits (B_{max}),

Quantization Step (q_s)

Output: Part2_3_length (P_{32})

Begin

1. $q_s = q_s + 1$
2. $\hat{I}$ = Quantizer(q_s, I)
3. *If* $\hat{I}_{max} > 8205$
   then goto Step 1
   Else goto Step 4.
4. P_{32} = HuffmanCoding($\hat{I}$)
5. $embedRule = (P_{32}\%2) \oplus (b)$
6. *If* $P_{32} > B_{max}$ or $embedRule = 1$
   then goto Step 1
   Else goto Step 7.
7. *return* P_{32}

End

Figure 3. Pseudo-code for inner loop function with MP3Stego

The inner loop quantizes the MDCT coefficients and increases the quantization step until the quantized coefficients can be coded with the available number of bits. The variable P_{32} contains the number of the bits used for scalefactors and Huffman code data for current granule, which is also called block length. Without embedding, the inner loop will be finished when the P_{32} is within the range of the number of bits available. In MP3Stego, the inner loop will continue to iterate until the parity of the P_{32} is equal to the hidden bit b and the bit demand for Huffman coding is met. Once the inner loop is done, another loop, namely outer loop, will check the distortions introduced by the quantization operation. If the allowed distortion is exceeded, the inner loop will be called again. The above process will be iterated until the bit rate and distortion requirements are both met.

III. Proposed Steganalytic Method

Obtaining the features that can reflect the differences between stego audios and cover audios is a crucial step for steganalysis. Since MP3Stego'information hiding takes place during the quantization process, it is natural and reasonable to generate feature from the parameters related to quantization. The parameter concerned in the proposed method is big_values in side information. In this section, the steganographic impact on big_values is analyzed first. The calibrated feature is obtained by re-compression in order to reduce the influence of the audio content. At last, we present the attacking algorithm for MP3Stego.

A. Steganographic Effect on Big_values

In order to improve the compression efficiency, lossless huffman coding is adopted in MP3 encoding. Quantized MDCT(QMDCT) coefficients is divided into three zones from high to low frequency: the all-zero region(Rzero), the small value region(Count1) and the large value region(Big_value) ,shown in Fig.4.

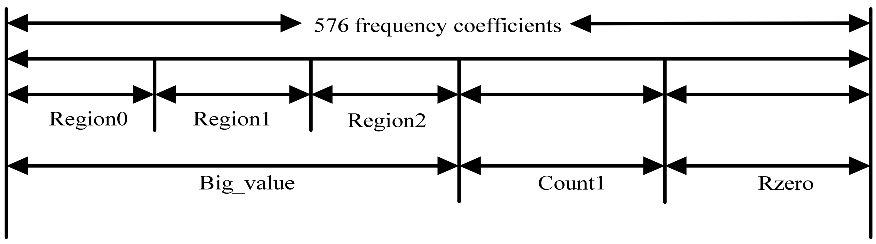


Figure 4. The partition of QMDCT coefficients

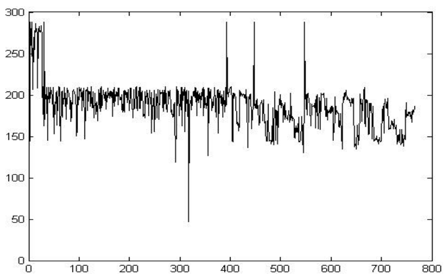
The Rzero region is not required to be encoded. In Count1 region, each of the four QMDCT coefficients is encoded. Each of the two QMDCT coefficients is encoded in Big_value region, which can be subdivided into Region0, Region1 and Region1 region. The number of QMDCT coefficients in Big_value, Count1 and Rzero region is represented by big_values ,count1 and rzero respectively. The value of big_values, count1 and rzero has following rules: if the value of two consecutive QMDCT coefficients is zero, then the value of rzero plus one; if the absolute value of four consecutive QMDCT coefficients is less than one, then the value of count1 plus one. As the value of big_values, count1 and rzero has the following condition:

$$2*rzero+4*count1+2*big\_values=576$$

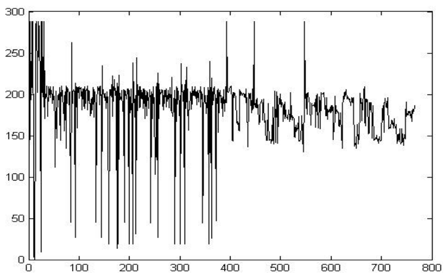
(6)

so the value of big_values can be obtained with rzero and count1.Because the relationship of rzero,count1 and big_values, the parameter of big_values is concerned in this article.

As described in Section2, it is obvious that more iterations are required to exit the inner loop due to the embedding. Consequently, we will have a larger quantization step at the end of the loop in stego case. The larger the quantization step is, the smaller the value of the QMDCT coefficients. So the value of big_values becomes small and the next frame's big_values becomes larger in stego case. Finally, the variance of big_values is larger than in non-stego case. This phenomenon is clearly seen in Fig.2 which is the distribution of big_values of the same MP3 audio in non-stego case and stego case. The horizontal axis represents the index of frame and the vertical axis represents the value of big_values. In stego case, the audio is embedded with 50% messages(100% corresponds to the maximum size of a message that can be embedded by MP3Stego).



(a)non-stego



(b)stego

Figure 5. Distribution of big_values in non-stego case and stego case

From Fig. 2, it is obvious to take the variance of big_values as feature to detect MP3Stego. That is

$$f=\frac{\sum_{i=1}^{N}\left(g_i-\bar{g}\right)^2}{N-1}$$

(7)

Where g_i ($i \in \{1,2,\cdots N\}$) is the value of big_values in i th granule and $\bar{g}$ is the mean value of big_values in all granules.

B. Calibration Feature by Re-compression

In steganalysis, it is almost impossible to have access to the cover during the process of steganalysis. If we can obtain the estimation of the cover from the suspect one as effectively as possible, the steganalystic performance will be improved. Several methods for cover estimation have been arisen in recent years, such as filtering [19], down-sampling [20], re-embedding [21], and re-compression [22]. One of the most famous approaches for creating an estimate of the cover image is the model proposed by Jessica Fridrich in [22] known as JPEG re-compression calibration. Similarly, we conjectured that the steganographic effect on big_values would be removed through re-compression calibration.

Denote M_d as an MP3 audio under scrutiny and M_e is the MP3 audio that M_d is decompressed to the time domain and the compressed back to MP3 with the same compression ratio. That is,

$$M_e=C\left(D\left(M_d\right),CR\right)$$

(8)

Where C and D denote the MP3 compression and decompression algorithm respectively, and CR is the compression ratio. Then we can obtain the final calibrated feature f, that is

$$f=\left|f_d-f_e\right|$$

(9)

Where f_d and f_e is the feature extracted from M_d and M_e respectively by equation (7).

C. Detecting MP3Stego

Denote M_d as an MP3 audio under scrutiny and it contains totally N granules. The process of detecting MP3Stego is described as follows.

Step1: Obtain the cover estimation M_e by decoding M_d and compressing again.

Step2: For the i th granule in M_d and M_e, obtain the value of big_values by the side information respectively.

Step3: Repeat Step 2 until reach the end. Finally, we can get the sequence of the value of big_values.

Step 4: Calculate f_d and f_e by equation (7) and then extract the feature $f = f_d - f_e$.

Step5: Take the f as input for SVM classifier. Apply SVM classifier and determine whether the testing MP3 audio has been deal with MP3Stego or not.

IV. Experimental Results

A. Experimental Setup

A total number of 500 mono audios with different styles are used and each audio is sampled at 44.1 kHz, 16 bits/sample and has the duration of 10s. They have never been MP3 compressed. Random sequence with 0 and 1 is taken as the secret message and no preprocessing such as compression or encryption is done to the message before embedding. All audios are embedded with 10%, 20%, 50%, 80%, 100% messages (100% corresponds to the maximum size of a message that can be embedded by MP3Stego).

In the experiments, half of the nature and stego audios are randomly selected to train the SVM classifier and the rest are used for testing.

B. Experimental Results and Discussions

In our experiments, true positive (TP) means that stego MP3 audio is predicted as stego MP3 audio, and true negative (TN) means that the nature MP3 audio is predicted as nature MP3 audio. Consequently, false negative (FN) means that the stego MP3 audio is predicted as nature MP3 audio, false positive (FP) means that the nature MP3 audio is predicted as stego MP3 audio. Since the stego MP3 audio and the nature MP3 audio to be detected have the same quantities in our experiments, the final detection accuracy rate (AR) is computed as $AR = (TPR + TNR)/2$, where $TPR = TP/(TP + FN)$,and $TNR = TN/(TN + FP)$. For each case, we average the ten times test results. Table 1 lists the average results under different bit-rates.

It is observed from Table I that the accuracy rate increases with the increase of embedding rate. It is in line with our expection because the more the information is hidden, the easier the audio after the operation of steganography will be detected.The detection results is good when the embedding rate is over 50%: all the results are over 90%. Especially when the embedding rate is less than 50%, the detection results are not good. This shows that the steganalysis of MP3Stego is still very challenging in cases with low embedding rates.

TABLE I.
AVERAGE RESULTS UNDER DIFFERENT BIT-RATES

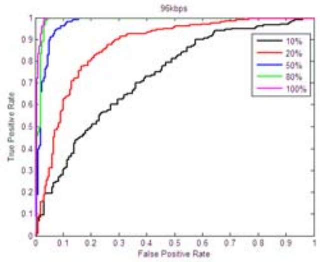
CR (kbps)	ER (%)	TPR(%)	TNR(%)	AR (%)
96	10	63.00	68.17	65.59
	20	73.92	83.92	78.92
	50	91.75	96.00	93.88
	80	94.92	99.42	97.17
	100	97.08	99.75	98.42
112	10	60.67	70.17	65.42
	20	73.42	80.17	76.80
	50	89.08	93.92	91.50
	80	93.92	97.42	95.67
	100	97.17	98.67	97.92
128	10	60.50	65.42	62.96
	20	67.58	76.17	71.88
	50	86.75	93.75	90.25
	80	93.67	95.33	94.50
	100	94.83	97.58	96.21

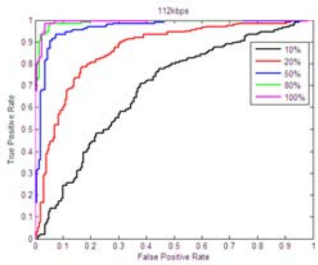
In order to show the detection reliability performance, receiver operating characteristic (ROC) curve has been used to verify the effectiveness of the proposed method. Fig. 6 gives the ROC curves under different compression ratios. The curves indicate that the proposed scheme is able to reliably detect the trace of MP3Stego. For instance, in Fig. 6(a), a probability of detection of approximately 95% is achieved at a false positive rate of 10%, when the embedding rate is 50%.

While the ROC curve contains most of the information about the accuracy of the classifier, it is sometimes desirable to produce quantitative summary measures of the ROC curve. The most commonly used such measure is the area under the ROC curves (AUC). In this paper, the detection reliability [22] is adopted to evaluate the detection performance, which is defined as,

$$\rho = 2A - 1 \tag{10}$$

where A is the AUC. A is scaled to obtain $\rho = 1$ for a perfect detection and $\rho = 0$ for a random guessing. The results for detection reliabilities under different embedding rates and different compression ratios are shown in Table II. The results in Table II show that the detection performance depends on the embedding rate. The less the embedding rate, the more difficult it is to detect.





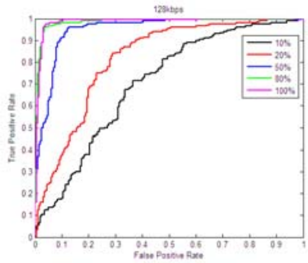


Figure 6. ROC curves under different embedding rates (a) 96kbps (b) 112kbps (c) 128kbps.

TABLE II.
AUC RESULTS UNDER DIFFERENT EMBEDDING RATES

CR (kbps)	ER (%)	AUC	ρ
96	10	0.7197	0.4394
	20	0.8661	0.7322
	50	0.9580	0.9160
	80	0.9908	0.9816
	100	0.9919	0.9838
112	10	0.7137	0.4274
	20	0.8505	0.7010
	50	0.9562	0.9124
	80	0.9768	0.9536
	100	0.9778	0.9556
128	10	0.6764	0.3528
	20	0.8194	0.6388
	50	0.9270	0.8540
	80	0.9762	0.9524
	100	0.9842	0.9684

V. CONCLUSIONS

In this paper, the effect on the big_values caused by MP3Stego embedding, is analyzed. The steganalytic features is obtained from the detecting and estimated audios obtained by re-compression calibration. The extraction of the feature is simple and rapid because they can be obtained directly from the MP3 bitstream without fully decoding. The experimental results show that the proposed method is effective to detect MP3Stego.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (NSFC: 60873220，61170137), Doctoral Fund of Ministry of Education of China(20103305110002), Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Y201119434), Key Innovation Team of Zhejiang Province (Grant No. C01416124200), The Outstanding (Postgraduate) Dissertation Growth Foundation of Ningbo University (Grant No. 10Y20100002),Zhejiang natural science foundation of China (ZJNSF:Z1090622,Y1090285), Zhejiang science & technology preferred projects of China(2010C11025)，Zhejiang province education department key project of China (ZD2009012) , Ningbo science & technology preferred projects of China(2009B10003), Ningbo key service professional education project of China(2010A610115).

REFERENCES

[1] S. Dumitrescu, X. Wu, Z. Wang, “Detection of LSB steganography via sample pair analysis,” IEEE Transactions on Signal Processing, 2003, vol. 51, pp. 1995-2007.

[2] N. Provos, P. Honeyman, “Hide and seek: an introduction to steganography,” IEEE Security & Privacy , 2003, vol.1, pp. 32-44.

[3] F. Petitcolas, MP3Stego. http://www.petitcolas.net/fabien/steganography/mp3stego, 2002.

[4] C. Platt, UnderMP3Cover. http://sourceforge.net/projects/ump3c, 2004.

[5] Z. Achmad, MP3Stegz. http://sourceforge.net/projects/mp3stegz, 2002.

[6] Noch, Stego-Lame. http://sourceforge.net/projects/stego-lame, 2002.

[7] D. Yan, R. Wang, L. Zhang, “Quantization step parity-based steganography for MP3 audio,” Fundamenta Informaticae, 2009, vol.97, pp. 1-14.

[8] D. Yan, R. Wang, “Huffman table swapping-based steganography for MP3 audio,” Multimedia Tools and Applications, 2011, vol. 52, pp. 291-305.

[9] A. Westfeld, “Detecting low embedding rates,” Proceedings of International Conference on Information Hiding, 2003, pp. 324-339.

[10] R. Bohme, A. Westfeld, “Statistical characterisation of MP3 encoders for steganalysis,” Proceedings of ACM Workshop on Multimedia and Security, 2004, pp. 25-34.

[11] J. Dittmann, D. Hesse, “Network based intrusion detection to detect steganographic communication channels: on the example of audio data,” Proceedings of IEEE International Workshop on Multimedia Signal Processing, 2004, pp 343-346.

[12] 12 C. Kraetzer, J. Dittman J, “Pros and cons of mel-cepstrum based audio steganalysis using SVM classification,” Proceedings of International Conference on Information Hiding, 2007, pp. 359-377.

[13] M. Qiao, A. Sung, Q. Liu, “Steganalysis of MP3Stego. Proceedings of International Joint Conference on Neural Networks,” 2009, pp. 2566-2571.

[14] M. Qiao, A. Sung, Q. Liu, “Feature mining and intelligent computing for MP3 steganalysis,” Proceedings of International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009, pp 627-630.

[15] H. Ozer, B. Sankur, N. Memon, I. Avcibas, “Detection of audio covert channels using statistical footprints of hidden messages,” Digital Signal Processing, 2006, vol.16, pp. 389-401.

[16] I. Avcibas, “Audio steganalysis with content-independent distortion measures,” IEEE Signal Processing Letters, 2006, vol. 13, pp. 92-95.

[17] ISO/IEC 11172-3: Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 M bit/s - part3: audio, 1993.

[18] 18 http://telos-systems.com/techtalk/hosted/Brandenburg_mp3_aac.pdf

[19] 19. AD. Ker, R. Bohme, “Revisiting weighted stego-image steganalysis,” Proceedings of SPIE on Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 2008, pp 5-17.

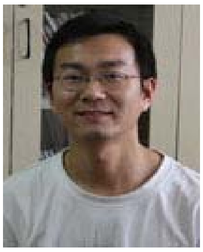
[20] 20. AD. Ker, “Steganalysis of LSB matching in grayscale images,” IEEE Signal Processing Letters, 2005, vol. 12, pp. 441-444.

[21] 21. H. Malik, “Steganalysis of qim steganography using irregularity measure,” Proceedings of ACM Workshop on Multimedia and Security, 2008, pp.149-158.

[22] 22. J. Fridrich J (2004) “Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes,” Proceedings of International Conference on Information Hiding, 2004, pp . 67-81.

Xianmin Yu is born in 1988. He is currently pursuing his M.S. degree in College of Information Science and Engineering, Ningbo University, China. His research interests mainly include multimedia retrieve and security.

Rangding Wang is born in 1962, received his M.S. degree in the Department of Computer Science and Engineering from the Northwest Polytechnic University, Xian in 1987, and received his Ph.D. degree in the School of Electronic and Information Engineering from Tongji University, Shanghai, China, in 2004. He is now a professor at Faculty of Information Science and Engineering, Ningbo University, China. His research interests mainly include multimedia security, digital watermarking for digital rights management, data hiding, and steganography for computer forensics.



Diqun Yan is born in 1979 and received his M.S. degree in circuit and system, in 2008, and his Ph.D. in communication and information system, in 2012, both from the College of Information Science and Engineering at the Ningbo University in China. His research interests include audio processing and multimedia security.