Adaptive MP3 Steganography Using Equal Length Entropy Codes Substitution

Kun Yang[1,2], Xiaowei Yi[1,2](✉), Xianfeng Zhao[1,2], and Linna Zhou[3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{yangkun9076,yixiaowei,zhaoxianfeng}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

[3] School of Information Science and Technology, University of International Relations, Beijing 100091, China
zhoulinna@mail.tsinghua.edu.cn

Abstract. Statistical undetectability is a common problem in current MP3 steganography. In this paper, a novel adaptive scheme of MP3 steganography is proposed for obtaining higher secure payload under the framework of distortion minimization. To avoid disabling MP3 encoder in the embedding process, a mapping construction algorithm using Huffman codes of equal length is realized to hold the length of stego codestream. Furthermore, a content-aware distortion function is designed to achieve optimal masking effect via the psychoacoustic model (PAM). Experimental results show that our method achieves better performance than others in terms of security and secure payload, the detection accuracy of the proposed method is lower than 55% under 128 kbps when the relative payload is equal to 0.25.

Keywords: MP3 steganography · Huffman code · Adaptive steganography · Distortion function · Psychoacoustic model

1 Introduction

The MP3 is becoming a kind of pervasive carrier for steganography at present, because it is very convenient to share speech or music in almost all public platform such as WeChat and YouTube. Many steganographic algorithms have been proposed in the audio compressed domain. However, they generally have low embedding capacity and very poor security. Conventional steganographic methods cannot satisfy security requirements, therefore adaptive MP3 steganography is inevitable tendency in modern steganography.

Recently, several steganographic methods have been proposed for MP3 audio files [1,5,7,8,11–15]. Wang et al. [11] presented a steganography algorithm which changes the position of the first nonzero value within the modified discrete cosine transform (MDCT) coefficients. In [12], according to the local signal-to-noise ratio, the feature vector of each frame is calculated for selecting embedding

C. Kraetzer et al. (Eds.): IWDW 2017, LNCS 10431, pp. 202–216, 2017.
DOI: 10.1007/978-3-319-64185-0_16

region, and then the secret massage is concealed via modifying MDCT coefficients. Liu et al. [7] proposed a steganography algorithm based on the energy of MDCT coefficients in adjacent frames. MP3Stego [8] is a well know MP3 steganographic method. The embedding operation is completed in the inner loop of MP3 encoder. The message bits are hidden based on the parity of the block length. Yan et al. [14] proposed a steganography algorithm by exploiting the rule of window switching. This algorithm establishes a mapping relationship between the secret bit and the encoding parameter, namely window type. Another algorithm proposed by Yan [13] is based on the parity of quantization step. The embedding operation is also accomplished in the inner loop of MP3 encoder. In addition, Yan [13] proposed a steganography algorithm based on the index of Huffman tables which used in MP3 encoding process. Gao et al. [5] and Yan et al. [15] respectively proposed algorithms based on Huffman codes mapping. The algorithms establish a mapping relationship between the secret bit and the Huffman code. Dong et al. [1] proposed two MP3 steganography algorithms by utilizing the Huffman codestream. Message bits are concealed within linbits or sign bits.

However, the above-mentioned steganographic methods generally have several disadvantages as follows: (1) Weak security. The detection accuracy is more than 80% with blind steganalysis [6]. (2) Low embedding capacity. For example, the maximum embedding capacity of the MP3Stego is about 154 bps (stereo, 44.1 kHz). (3) Non-adaptivity. Distortion minimization is not obeyed in existing methods. Adaptive steganography is the state-of-the-art technique to overcome the shortages [3], which is verified primely in the field of adaptive image steganography. Due to the different between the human auditory system (HAS) and the human visual system (HVS), the adaptive image steganography cannot be used directly for the audio steganography. Therefore, an adaptive MP3 steganography method is proposed in this paper, which is compatible with distortion minimization framework. In the proposed method, a mapping construction is established between Huffman codes of equal length and the binary bitstream. In addition, a distortion function is defined based on psychoacoustic model (PAM) for achieving optimized imperceptibility and security. The message bits are embedded within the binary bitstream using the syndrome-trellis codes (STCs).

The rest of this paper is organized as follows. Section 2 introduces MP3 encoding standard briefly. An adaptive steganographic scheme is proposed for the MP3 files in Sect. 3. Experimental results and discussion are shown in Sect. 4. Finally, conclusion is drawn in Sect. 5.

2 Overview of MP3 Encoder

The procedure of the MP3 encoding is shown in Fig. 1, which can be mainly divided into six steps [9]: framing, subband filter, PAM, MDCT, quantization, and Huffman encoding. The original audio is firstly partitioned into frames of 1152 samples in framing process, and each frame is further split into two granules for encoding independently. Filter bank which consists of 32 channels divides the

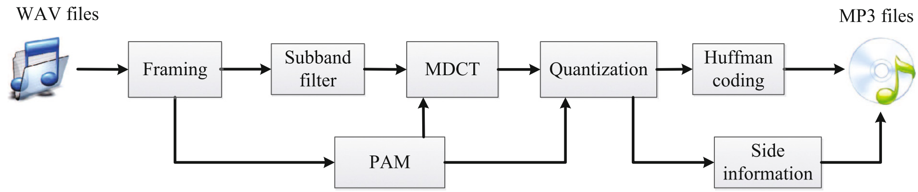


Fig. 1. The architecture of the MP3 encoder.

audio signal into 32 uniformly spaced frequency subbands. The MDCT process performs time-frequency transform. PAM process analyzes the audio signal and computes the perceptual entropy (PE). After time-frequency transform, encoder performs non-uniform quantization of the frequency lines. The quantized MDCT (QMDCT) coefficients are encoded by Huffman encoding process. Then Huffman codes and the side information are formatted into MP3 file.

The PAM is mainly based on the physical structure and the perception model of the HAS. The main purpose of PAM is to determine the window type based on the PE, which is used in the MDCT process and bit allocation. On the basis of the PAM, encoder removes the redundant information of the audio signals for efficient compressing. Including the noise, some high-frequency components of the signal are less than the auditory threshold or the masking threshold, which cannot be perceptual by human.

Quantization is the process of constraining an input from a continuous set of values to a discrete set. The key steps of quantization are realized in a three-layer iterative loop including frame loop, outer loop, and inner loop. The frame loop is designed to complete parameters initialization and to calculate remaining bits of the bit-reservoir. The outer loop calculates the quantization distortion of each scale factor band and estimates whether the distortion is under the masking threshold. Quantization is accomplished in inner loop by constantly adjusting the quantization step. Furthermore, the inner loop must meet two conditions: (1) All of QMDCT coefficients are no bigger than the theoretical maximum in the MP3 standard. (2) The length of Huffman codestream of quantized coefficients is less than the available bits.

As described in Fig. 2, a granule consist of 576 QMDCTs, which is orderly divided into three kinds of regions: big-value region, count1 region, and rzero region. Since the values of QMDCTs are equal to 0 within the rzero region, all these coefficients are no longer encoded. The values of QMDCTs in count1 region belong to $\{\pm 1, 0\}$, and quaternate coefficients are encoded by using Huffman code Table 32 and Table 33. Different with the count1 region, for the big-value region, each pair of QMDCTs are encoded using a Huffman code. The big-value region is further divided into region0, region1, and region2. Three subregions are encoded independently by utilizing the Huffman tables from Table 0 to Table 31 (Table 4 and Table 14 are unusable). If the value of a QMDCT is less than 15 then it is coded directly, otherwise the exceeding value is represented using linbits. Each nonzero coefficient possesses a sign bit, 0 indicates positive number

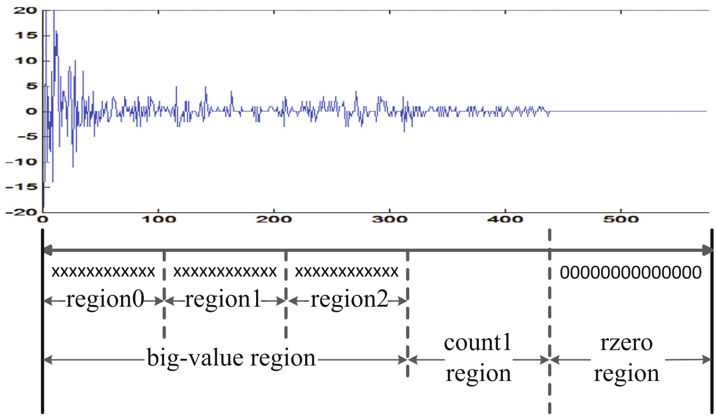


Fig. 2. Organization of QMDCTs within a granule.

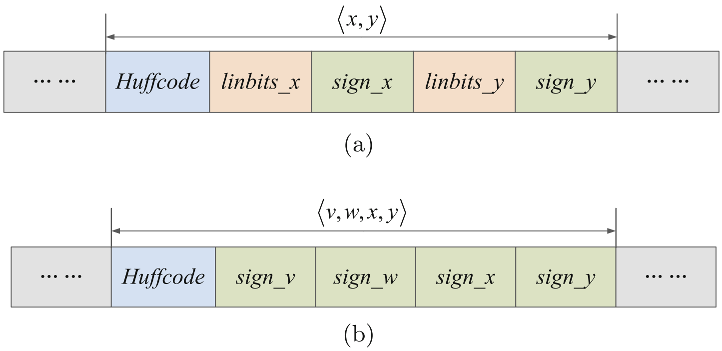


Fig. 3. Codestream structure: (a) Big-value region, and (b) Count1 region.

and 1 indicates negative number. The bitstream after Huffman coding is shown in Fig. 3. *Huffcode* denotes a Huffman code of QMDCTs, $linbits\_x$ denotes the linbits of the first coefficient, $sign\_x$ denotes the sign bit of the first coefficient, $linbits\_y$ denotes the linbits of the second coefficient, $sign\_y$ denotes the sign bit of the second coefficient. Likewise, $sign\_v$, $sign\_w$, $sign\_x$, and $sign\_y$ sequentially denotes the sign bits of four QMDCTs.

3 Proposed Steganography Method

A novel adaptive steganography algorithm based on equal length entropy codes substitution (EECS) is proposed. The embedding process is located between the quantization process and the Huffman encoding process in MP3 encoder. As shown in Fig. 4, the proposed steganographic algorithm is mainly divided into four steps: code-to-binary process, distortion function computing, STCs process, binary-to-code process. First, the cover Huffman code stream H_c is transformed

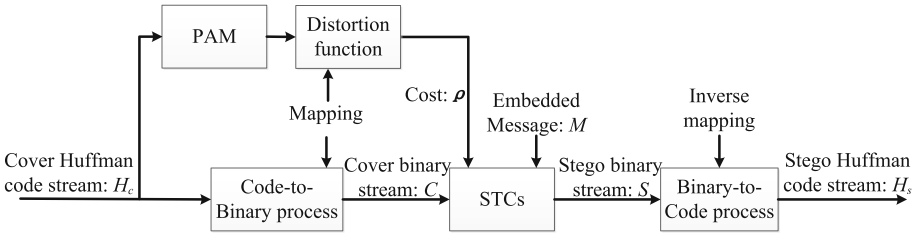


Fig. 4. A flowchart of embedding process of the proposed adaptive steganography.

into the cover binary stream C in the code-to-binary process. Then the distortion ρ is calculated based on the PAM and the mapping distance of Huffman codes. After that, message bits M are concealed in C by STCs [2]. Finally, the stego binary stream S is similarly inverse mapped to the stego Huffman code stream H_s in the binary-to-code process. Therefore, the embedding algorithm is generalized modeling as follows:

$$STC(C, M, \rho) = S, \tag{1}$$

$$H_c \times S \rightarrow H_s. \tag{2}$$

3.1 Code Substitution Construction

In the proposed EECS, Huffman codes in the big-value region are used to embed message. For holding the codestream length and the statistical distribution of Huffman codes, so the structure of a coding unit described in Fig. 3 cannot be altered. However, the Huffman code length is varied because of variable-length entropy encoding. Suppose that $h_i^{(k)}$ denotes the ith *Huffcode* within the kth Huffman table, $\langle x_i^{(k)}, y_i^{(k)} \rangle$ denotes the corresponding QMDCTs pair of $h_i^{(k)}$. Thus for $\forall i \neq j$, $h_i^{(k)}$ and $h_j^{(k)}$ is a pair of substitutable codes, which satisfies the following three conditions:

(C1) Length of Huffman codes. The length of $h_i^{(k)}$ is equal to the length of $h_j^{(k)}$,

$$L(h_i^{(k)}) = L(h_j^{(k)}). \tag{3}$$

(C2) Number of sign bits. The number of sign bits of $\langle x_i^{(k)}, y_i^{(k)} \rangle$ is equal to the number of sign bits of $\langle x_j^{(k)}, y_j^{(k)} \rangle$,

$$\Theta(\langle x_i^{(k)}, y_i^{(k)} \rangle) = \Theta(\langle x_j^{(k)}, y_j^{(k)} \rangle). \tag{4}$$

(C3) Linbits flags. The linbits flag of $x_i^{(k)}$ is consistent with the linbits flag of $x_j^{(k)}$, likewise the same as $y_i^{(k)}$ and $y_j^{(k)}$,

$$G(x_i^{(k)}) = G(x_i^{(k)}), G(y_i^{(k)}) = G(y_i^{(k)}). \tag{5}$$

According to (C3) in (5), a conservative scheme is just to use QMDCTs whose values are less than 15 in big-value region for hiding message. Based on the above-mentioned conditions in (3)–(5), a code substitution construction is described in detail as follows.

The set $\varPi^{(k)}$ contains all Huffman codes in the kth Huffman table. In the proposed algorithm, $\varPi^{(k)}$ is divided into two subsets: $\varPi_u^{(k)}$ and $\varPi_v^{(k)}$. Huffman codes in $\varPi_v^{(k)}$ are available for embedding message. $\varPi_v^{(k)}$ is generated via a iterator. $\varPi_v^{(k)}$ is initialized to $\varnothing$. After that, for $\exists h_i, h_j \in \varPi^{(k)} \backslash \varPi_v^{(k)} (i \neq j)$, if (h_i, h_j) satisfies the conditions in (3)–(5), then moving h_i and h_j to $\varPi_v^{(k)}$, otherwise moving them to $\varPi_u^{(k)}$. Repeating this process until $\varPi^{(k)} = \varnothing$.

Each Huffman code h_i in $\varPi_v^{(k)}$ is numbered according to the placement order of putting into $\varPi_v^{(k)}$. Huffman codes in $\varPi_v^{(k)}$ are divided into $\varPi_0^{(k)}$ and $\varPi_1^{(k)}$. When the placement order is odd, h_i is put into $\varPi_1^{(k)}$, otherwise it is put into $\varPi_0^{(k)}$. Huffman codes in $\varPi_0^{(k)}$ represent the bit '0'. Contrarily, Huffman codes in $\varPi_1^{(k)}$ represent the bit '1'. According to the distribution characteristics of Huffman codes length, two substitutable Huffman codes are generally more closer with zigzag scanning. As an example, the traversal of the No. 7 Huffman table is described in Fig. 5. Based on (C2) in (4), QMDCT pairs are classified into R_0, R_1, and R_2, which are sealed for independently performing search iterators. The traversal order is indicated by arrows in Fig. 5. These Huffman codes displayed with solid points are $h_{\langle 2,3\rangle}, h_{\langle 3,2\rangle}, h_{\langle 5,1\rangle}, h_{\langle 4,2\rangle}, h_{\langle 2,4\rangle}, h_{\langle 1,5\rangle}$. The length of these six Huffman codes is equal to eight. According to the traversal order, these six Huffman codes constitute three mapping pairs: $h_{\langle 2,3\rangle} \longleftrightarrow h_{\langle 3,2\rangle}$, $h_{\langle 5,1\rangle} \longleftrightarrow h_{\langle 4,2\rangle}$, $h_{\langle 2,4\rangle} \longleftrightarrow h_{\langle 1,5\rangle}$. As described in Fig. 6, $h_{\langle 2,3\rangle}, h_{\langle 5,1\rangle}, h_{\langle 2,4\rangle}$ are in $\varPi_1^{(k)}$,

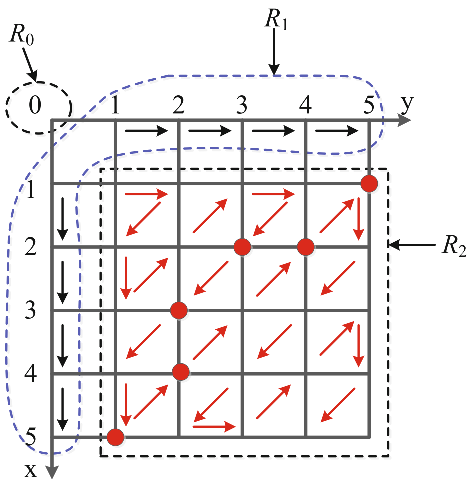


Fig. 5. Zigzag scanning of the #7 Huffman table.

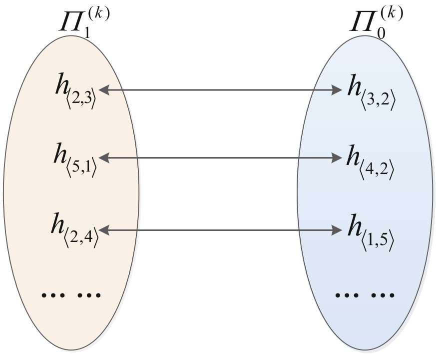


Fig. 6. An example of mapping relations of substitutable Huffman codes.

and $h_{\langle 3,2\rangle}$, $h_{\langle 4,2\rangle}$, $h_{\langle 1,5\rangle}$ are in $\Pi_0^{(k)}$. Hence, in the code-to-binary process, the transfer function is defined as:

$$f_{ctb}(h) = \begin{cases} 0, & \text{if } h \in \Pi_0^{(k)}, \\ 1, & \text{if } h \in \Pi_1^{(k)}, \\ \varnothing, & \text{otherwise.} \end{cases} \tag{6}$$

Likewise, the inverse process is presented as:

$$f_{btc}(m, g) = \begin{cases} g, & \text{if } m = 0, g \in \Pi_0^{(k)}, \\ \hat{g}, & \text{if } m = 0, g \in \Pi_1^{(k)}, \\ \hat{g}, & \text{if } m = 1, g \in \Pi_0^{(k)}, \\ g, & \text{if } m = 1, g \in \Pi_1^{(k)}, \end{cases} \tag{7}$$

where $(g, \hat{g})$ is a mapping pair of substitutable Huffman codes.

3.2 Distortion Function

Suppose that the distortion which results from substitution of each Huffman code is mutually independent in the proposed algorithm. As described in Filler's work [2], the total distortion is modeling as follows:

$$D(X, Y) = \sum_{i=1}^{n} \rho_i(h_i, h_i^{'}), \tag{8}$$

where $\rho_i(h_i, h_i^{'})$ is the distortion of mapping between Huffman code h_i and $h_i^{'}$, n is the number of the Huffman codes which are modified in the embedding process.

The distortion in the proposed algorithm is mainly affected by two factors: the influence of PAM and the modified magnitude of coefficients. These two factors cause the degeneration of the audio and the change in statistical characteristics of Huffman codes.

The influence of PAM is mainly reflected as follows. The hearing frequency range of the human ear is usually between 20 Hz and 20 kHz. However, the human ear has different sensitivity to each frequency band. The sensitivity can be reflected by the absolute threshold of hearing which is obtained by large number of experiments. The absolute threshold of hearing describes how much energy a pure tone needs to be heard in a noise free environment at different frequencies. It can be approximated by the following formula:

$$T_f = 3.64 \times \left(\frac{f}{1000}\right)^{-0.8} - 6.5 \times \mathrm{e}^{-0.6\times(\frac{f}{1000}-3.3)^2} + 10^{-3} \times \left(\frac{f}{1000}\right)^4, \quad (9)$$

where f is the frequency, T_f is the absolute threshold of hearing at frequency f. The smaller T_f means the higher sensitivity of human ear to audio signal. It is more likely to cause the distortion when the modification occurs in the more sensitive frequency band.

The modified magnitude of coefficients also affects the distortion. When Huffman code h_i is replaced by $h_i^{'}$, the QMDCT coefficients pair $\langle x_i, y_i\rangle$ becomes $\langle x_i^{'}, y_i^{'}\rangle$ correspondingly. This change leads to the degeneration of audio. To measure the distortion, the difference between $\langle x_i, y_i\rangle$ and $\langle x_i^{'}, y_i^{'}\rangle$ is calculated firstly. The definition of difference is given by:

$$d_i = \left|x_i^{'} - x_i\right| + \left|y_i^{'} - y_i\right|, \quad (10)$$

where d_i is the difference between h_i and $h_i^{'}$. The greater the d_i, the larger the difference between these two Huffman codes.

From the above analysis, the distortion function is defined as:

$$\rho_i = \frac{1}{\log_2(\frac{t_{2i}+t_{2i+1}}{2} + \sigma)} \times d_i = \frac{\left|x_i^{'} - x_i\right| + \left|y_i^{'} - y_i\right|}{\log_2(\frac{t_{2i}+t_{2i+1}}{2} + \sigma)}, \quad (11)$$

where i is the index of h_i in the Huffman code stream, t_{2i} is the absolute threshold of hearing of the $(2i)$th frequency line, i.e. the T_f when the f is the $(2i)$th frequency line, σ is a constant to ensure that $(\frac{t_{2i}+t_{2i+1}}{2} + \sigma)$ is greater than 0, the logarithmic operation reduces the influence of some extreme values.

3.3 Embedding Procedure and Extraction Procedure

Embedding Procedure. First, the cover Huffman code stream H_c is transformed into cover binary stream C. $D_{df}(\cdot)$ is the distortion function. Then C and the distortion ρ are scrambled using the same key k. Message bits M are embedded into C by STCs. Finally, the stego binary stream S is transformed into stego Huffman code stream H_s after the inverse operation of scrambling. The pseudocode of embedding procedure is described as Algorithm 1.

Algorithm 1. The process of information embedding

Input: H_c, M, k, σ;
Output: H_s;
for $i = 1$ to the length of H_c **do**
    $C[i] = f_{ctb}(H_c[i])$
end for
$\rho = D_{df}(H_c, \sigma)$
Scramble the C and ρ using the key k
$S = STC(C, M, \rho)$
Inverse scramble the S
for $i = 1$ to the length of H_c **do**
    $S[i] = f_{btc}(S[i], H_c[i])$
end for
return H_s

Extraction Procedure. First, the stego Huffman code stream H_s is transformed into stego binary stream S. Then S is scrambled using the same key as embedding process. Finally, secret message bits M are extracted from S by STCs. The pseudocode of extraction procedure is described as Algorithm 2.

Algorithm 2. The process of information extraction

Input: H_s, key
Output: M
for $i = 1$ to the length of H_s **do**
    $S[i] = f_{ctb}(H_s[i])$
end for
Scramble S using the same key k with embedding process
Extract M from S by $STCs$
return M

4 Experimental Results

The experimental settings are shown in Table 1 in detail, where w and h are the width and the height of the generated matrix in STCs, respectively. And σ is the parameter in (11). Our experiments evaluate the proposed algorithm in three aspects: embedding capacity, imperceptibility, undetectability. Meanwhile, the proposed algorithm is compared with the other three steganography algorithms. These three algorithms are MP3Stego [8], Huffman Code Mapping (HCM) [15] and Adaptive Post-Steganography (APS) [16]. The MP3Stego is the first audio steganography software for MP3. Meanwhile, it is a classical audio steganography algorithm. The HCM is a steganography algorithm which is also based on Huffman code mapping. The APS embeds message in count1 region rather than big-value region.

Table 1. The settings of experimental parameters

	Parameter	Value
Datasets	Quantity	1000
	Channel	Stereo
	Genres	Blues, classical, country, folk, pop, jazz
	Duration	10 s
	Digitalizing bit	16 bits
LAME	Version	Lame-3.99.5
	Bit rate	128 kbps, 320 kbps
	Other parameters	Default
EECS	w	2, 4, 10
	h	7
	σ	10

4.1 Embedding Capacity

The embedding capacities of these four algorithms under various bit rates and w are shown in Fig. 7.

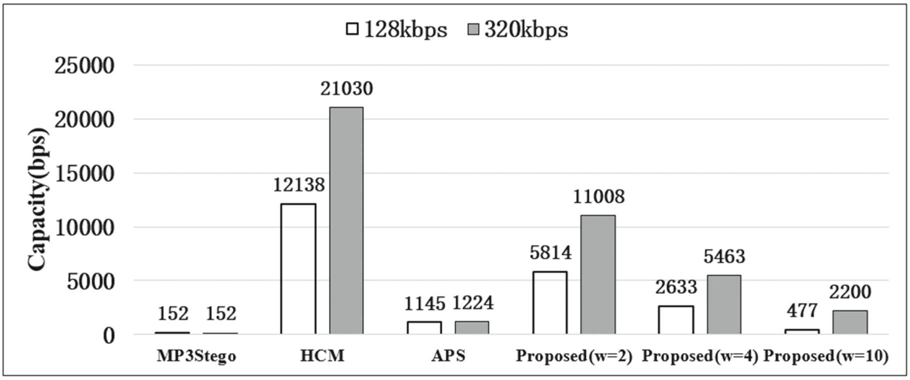


Fig. 7. Comparison of the embedding capacity.

From the Fig. 7, we can see that the embedding capacity of EECS is higher than that of MP3Stego but lower than that of HCM. That is because the payload of EECS is controlled by w and the maximum payload is 0.5 when w is equal to 2. Therefore, the embedding capacity of EECS (w is equal to 2) is roughly half that of HCM. The embedding capacity of EECS is higher than that of APS except one situation that the bit rate is 128 kbps and w is equal to 10. The greater the w, the lower the payload. This situation can be reflected in the Fig. 7

that great w results in the low embedding capacity. In addition, the embedding capacity of EECS increase with increasing the bit rate. When the bit rate is set to 320 kbps and w is equal to 2, the embedding capacity of EECS is up to 11 kbps.

4.2 Imperceptibility

Perceptual evaluation of audio quality (PEAQ) [10] is a standardized algorithm for objectively measuring perceived audio quality. The objective difference grade (ODG) is the main output parameter of this perceptual measurement method. Generally, the ODG has a range between 0 and −4. The closer the ODG value is to 0, the higher the similarity between the test audio and the reference audio. When the similarity of two audios is very high, the ODG value may be greater than 0. In this experiment, the reference audio is the WAV decompressed from cover MP3 and the test audio is the WAV decompressed from stego MP3. In addition, the imperceptibility is influenced by the embedding rate (ER). The ER here is defined as:

$$ER = \frac{n}{N}, \tag{12}$$

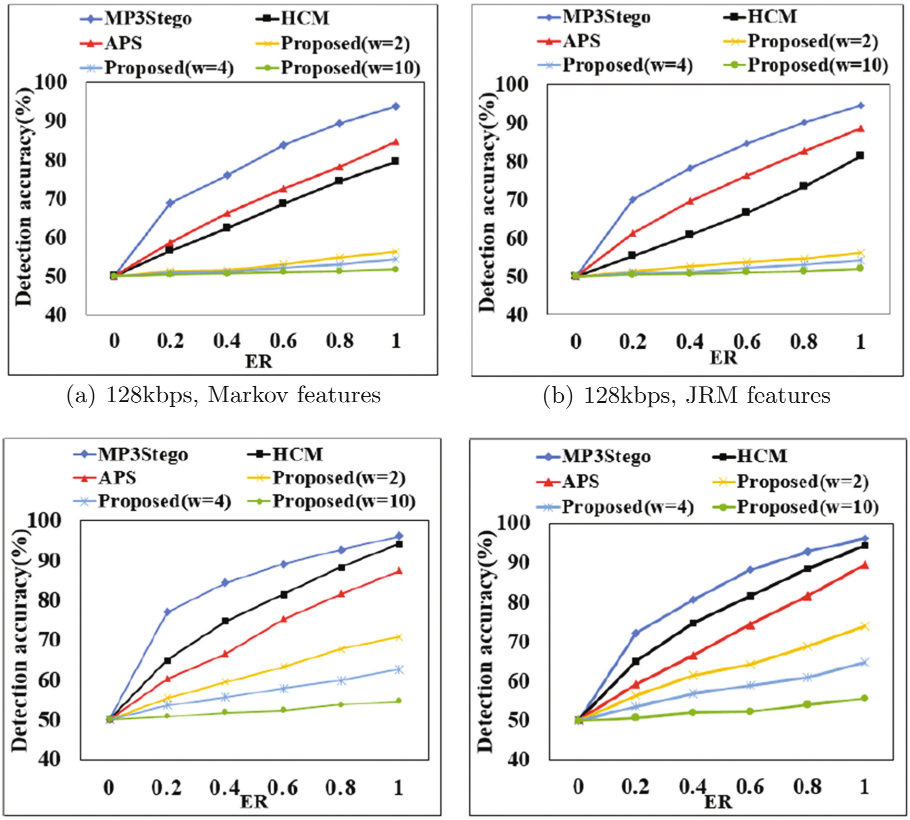
where N is the total number of frames of the audio file, n is the number of frames used to embed message. The results are shown in Tables 2 and 3.

Table 2. The ODG values at bit rate = 128 kbps

	ER				
	0.2	0.4	0.6	0.8	1.0
MP3Stego	−0.569	−0.711	−0.854	−0.997	−1.105
HCM	−0.551	−0.684	−0.812	−0.914	−1.013
APS	−0.548	−0.672	−0.801	−0.906	−1.004
Proposed (w = 2)	−0.473	−0.535	−0.627	−0.701	−0.796
Proposed (w = 4)	−0.445	−0.487	−0.536	−0.573	−0.611
Proposed (w = 10)	−0.427	−0.442	−0.479	−0.492	−0.513

Table 3. The ODG values at bit rate = 320 kbps

	ER				
	0.2	0.4	0.6	0.8	1.0
MP3Stego	0.043	0.028	0.014	−0.002	−0.011
HCM	0.048	0.041	0.032	0.024	0.013
APS	0.042	0.033	0.021	0.013	0.005
Proposed (w = 2)	0.051	0.042	0.038	0.030	0.022
Proposed (w = 4)	0.053	0.048	0.042	0.038	0.034
Proposed (w = 10)	0.056	0.054	0.051	0.049	0.047



(a) 128kbps, Markov features

(b) 128kbps, JRM features

(c) 320kbps, Markov features

(d) 320kbps, JRM features

Fig. 8. Detection accuracy using Markov and JRM features

It is obvious that the ODG value of EECS is greater than that of MP3Stego HCM and APS all the time. When the bit rate is set to 128 kbps, the ODG values of these algorithms are all lower than 0. On the contrary, when the bit rate is 320 kbps, the ODG values are basically greater than 0. The ODG values decrease with increasing the ER. Besides, the ODG value of EECS increases with increasing the parameter w. Thereby it proves that the proposed algorithm has good performance on imperceptibility.

4.3 Undetectability

In the experiments, we use steganalysis algorithms which are proposed in [4,6] to detect the proposed algorithm and the other three algorithms. The steganalysis algorithm proposed in [6] is based on Markov features. Markov features are often used in audio steganalysis and achieve good results. And the algorithm proposed in [4] is based on the JPEG domain rich model (JRM). JRM features are outstanding discrete cosine transform domain features to detect steganography. In the experiments, 70% of the samples are used for training and the others are used for testing. The classifier used in our experiment is support vector machine. The lower the detection accuracy, the higher the security of steganography algorithm. The results are shown in Fig. 8.

It is obvious that for any value of the bit rate and w, the detection accuracy of the proposed algorithm is the lowest. The detection accuracy of the proposed method is far lower than the other three algorithms especially when the bit rate is set to 128 kbps. In addition, the detection accuracy increases with increasing the bit rate and increases with decreasing the parameter w. That is because the high embedding capacity results in the low security. When the embedding capacity of EECS reaches the maximum, the detection accuracy is still lower than 75%. A conclusion can be made that the EECS has better undetectability than the other three steganography algorithms.

Another method to verify the security of the algorithm is analyzing the statistical distribution of Huffman codes. This method detects the steganography algorithm by calculating the similarity of Huffman codes distribution between the cover and the stego. The similarity is measured using the conditional entropy of two sequences. The smaller the conditional entropy, the higher the similarity.

Table 4. The conditional entropy measured at ER = 1, bit rate = 128 kbps

	Table No.					
	9	12	15	24	25	26
MP3Stego	0.98254	1.31427	1.78192	1.88619	1.60931	1.16023
HCM	0.00295	0.01962	0.03302	0.01987	0.01002	0.00411
APS	0.00621	0.00932	0.01448	0.00512	0.00603	0.00404
Proposed (w = 2)	0.00169	0.00401	0.00425	0.00493	0.00311	0.00262
Proposed (w = 4)	0.00037	0.00116	0.00139	0.00174	0.00122	0.00098
Proposed (w = 10)	0.00003	0.00002	0.00011	0.00015	0.00006	0.00004

Table 5. The conditional entropy measured at ER = 1, bit rate = 320 kbps

	Table No.					
	9	12	15	24	25	26
MP3Stego	1.14861	1.36296	1.94305	1.96382	1.73284	1.28427
HCM	0.09721	0.05236	0.06326	0.03908	0.02017	0.00932
APS	0.00701	0.01039	0.01593	0.00869	0.00827	0.00502
Proposed (w = 2)	0.00425	0.00721	0.00983	0.00924	0.00615	0.00631
Proposed (w = 4)	0.00092	0.00322	0.00417	0.00472	0.00326	0.00202
Proposed (w = 10)	0.00037	0.00031	0.00084	0.00072	0.00033	0.00029

And the higher the similarity, the better the undetectability. The function of calculating the similarity is defined as follows:

$$H_p(q) = \sum_{i=1}^{n} q_i \log_2 \left(\frac{q_i}{p_i} \right), \tag{13}$$

where n is the number of bins in the distribution histogram of Huffman codes, p_i is the probability of the ith bin in the distribution histogram of Huffman codes in cover, q_i is the probability of the ith bin in the distribution histogram of Huffman codes in stego. Six Huffman tables, which have higher usage frequency than others, are used in our experiments. The other Huffman tables have similar results. The experimental results are shown in Tables 4 and 5.

From the experimental results, it is obvious that the conditional entropies of EECS are lower than other algorithms. It can be inferred that the Huffman codes distribution is kept well after embedding message in the audio. This result is consistent with the result of the steganalysis algorithms above. The smaller the change of Huffman code distribution histogram, the lower the detection accuracy. Therefore, it can be concluded that the proposed algorithm has better undetectability.

5 Conclusion and Future Work

In this paper, we propose an adaptive MP3 steganography algorithm. In the proposed algorithm, a mapping relationship between Huffman codes and binary bits is established. In addition, a content-aware distortion function based on PAM and the mapping is constructed. The message is embedded in the audio file by substituting the Huffman codes of equal length. Experimental results show that the proposed algorithm does better than the other three audio steganography algorithms in terms of security and secure payload. The detection accuracy of the proposed method is lower than 55% under 128 kbps when the relative payload is equal to 0.25.

However, a shortcoming of the proposed method is that it doesn't take into account the statistical characteristics of distribution when the cost is calculated. For example, the statistical distribution of Huffman codes isn't taken into account. Although the experiment demonstrates that the Huffman code distribution is kept well in the proposed method, it is still a weakness for the security of the proposed algorithm. In the future work, we will construct a distortion function based on other aspects of PAM or statistical characteristics of Huffman code and achieve steganography based on the multiple-base system.

Acknowledgments. This work was supported by the NSFC under U1636102, U1536207 and U1536105, and National Key Technology R&D Program under 2014BAH41B01, 2016YFB0801003, 2016QY15Z2500 and 2016QY08D1600.

References

1. Dong, Y.: Research on information hiding based on mp3. Master's thesis, Beijing University of Posts and Telecommunications (BUPT), Beijing (2015)
2. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Trans. Inf. Forensics Secur. **6**(3), 920–935 (2011)
3. Fridrich, J.: Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, New York (2009)
4. Fridrich, J.: Steganalysis of JPEG images using rich models. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 8303, pp. 7–20 (2012)
5. Gao, H.: The mp3 steganography algorithm based on huffman coding. Acta Scientiarum Naturalium Univ. Sunyatseni **46**(4), 32–35 (2007)
6. Jin, C., Wang, R., Yan, D.: Steganalysis of mp3stego with low embedding-rate using markov feature. Multimed. Tools Appl. **76**(5), 6143–6158 (2017)
7. Liu, W., Wang, S., Zhang, X.: Audio watermarking based on partial mp3 encoding. Acta Scientiarum Naturalium Univ. Sunyatseni **43**(S2), 26–33 (2004)
8. Petitcolas, F.A.: MP3Stego (1998)
9. Smith, J.O., Abel, J.S.: ISO/IEC 11172–3: Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1.5 Mbit/s - Part 3: Audio (1993)
10. Thiede, T.: PEAQ-the ITU standard for objective measurement of perceived audio quality. J. Audio Eng. Soc. Audio Eng. Soc. **48**(1), 3–29 (2000)
11. Wang, C., Chen, T., Chao, W.: A new audio watermarking based on modified discrete cosine transform of MPEG/audio layer III. In: Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, pp. 984–989 (2004)
12. Wu, G., Zhuang, Y., Wu, F., Pan, Y.: Adaptive audio watermarking based on snr in localized regions. J. Zhejiang Univ. Sci. A (Sci. Eng.) **6A**, 53–57 (2005)
13. Yan, D.: Steganography and Steganalysis of Compressed Domain Audio. Ph.D. thesis, Ningbo University, Ningbo (2012)
14. Yan, D., Wang, R., Yu, X., Zhu, J.: Steganography for MP3 audio by exploiting the rule of window switching. Comput. Secur. **31**(5), 704–716 (2012)
15. Yan, D., Wang, R., Zhang, L.: A large capacity MP3 steganography algorithm based on huffman coding. J. Sichuan Univ. (Nat. Sci. Ed.) **48**(6), 1281–1286 (2011)
16. Zhang, Y., Pan, F., Shen, J., Li, N.: A post-adaptive steganography algorithm based on MP3. Comput. Sci. **43**(8), 114–117 (2016)