doi:103969/j. issn. 0490-6756. 2011. 06. 011

基于 Huffman 编码的大容量 MP3 隐写算法

严迪群，王让定，张力光
(宁波大学信息科学与工程学院，宁波 315211)

摘 要：本文针对 MP3 编码标准中哈夫曼码字对特点，提出了一种借助码字替换实现秘密信息隐写的新算法. 该算法首先对哈夫曼码表中的码字进行分类，以保证替换操作不改变 MP3 码流的固定结构，再借鉴混合进制的概念，采用多进制方式隐藏秘密信息. 给出了算法在二进制和多进制两种模式下的仿真结果，表明多进制隐写模式可以获得更高的隐写速率和效率，同时算法的感知透明性也能得到较好保持.

关键词：哈夫曼编码；混合进制；隐写
中图分类号：TP391 **文献标识码**：A **文章编号**：0490-6756(2011)06-1281-06

A high capacity MP3 steganography based on Huffman coding

YAN Di-Qun, WANG Rang-Ding, ZHANG Li-Guang
(College of Information Science and Engineering, Ningbo University, Ningbo 315211, China)

Abstract: A high capacity steganography method for mp3 audios is proposed in this paper. According to the characteristic of Huffman coding, the code words in Huffman tables are first classified to ensure that the embedding operation does not change the bitstream structure in MP3 standard. Then secret data are embedded by replacing the corresponding code words. The embedding strategy is based on multiple-base nation system. The structure of bit stream and the size of the cover audio can be kept unchanged after embedding. The results show that the proposed method can obtain higher hiding capacity and better efficiency than that of the method under binary case. Furthermore, the imperceptibility can also be better maintained in our method.

Key words: Huffman coding, multi-base nation, steganography

1 引 言

所谓隐写，是指以数字多媒体作品为掩护载体，把秘密信息隐藏到载体信号中，以不引起外界注意的方式通过公开信道进行传输的过程. 之所以能够在数字多媒体作品中实现隐写，主要是因为作品本身存在冗余，而这些冗余可以用秘密信息取而代之. 就数字音频作品的隐写技术方面，目前更多的研究还是集中在非压缩音频格式上(典型的如 WAV 音频)，而且也已出现了很多成熟的方法和工具[1-3]. 但众所周知，音频作品在未经压缩之前其数据量是非常大的，庞大的数据量将对作品的存储和传输造成极大不便，因此无论是在互联网还是其它传输信道，绝大多数的音频作品是以压缩编码的格式出现的. 对于隐写系统的攻击者而言，压缩格式音频所具有这种的广泛性和普遍性将对其造成极大的迷惑. 另一方面，压缩音频作品的大小一般在 3～5 M，相对于其他类型的载体(如图片、文

收稿日期：2010-12-24
基金项目：国家自然科学基金(60873220,61170137)；宁波大学科研基金(XYL10002,XK1087)
作者简介：严迪群(1979—)，男，浙江省余姚人，博士研究生，主要研究方向为信息隐藏等. E-mail: yandiqun@nbu. edu. cn

本等)更容易实现大量秘密信息的隐写操作. 因此,研究压缩格式下音频作品的隐写技术具有更强的现实意义.

一般来讲,针对压缩格式音频的信息隐藏算法可分为以下三类.

(1) 利用MPEG量化编码的内循环终止条件来隐藏秘密信息,如MP3Stego[4],其通过调节量化误差,将量化编码后块长度的奇偶性作为数据隐写的依据,但这类方法鲁棒性较差,任何攻击者可通过解压比特流再重新压缩,去除隐藏的秘密信息,同时隐写速率也很低. 另一方面也有研究[5,6]指出,通过分析MP3音频数据块长度的统计特性可检测出该载体是否经过MP3Stego算法的处理,由此可见这类算法的安全性有待进一步提高.

(2) 在MPEG编码过程中通过修改MDCT系数来实现秘密信息的隐写. Wang[7]通过大量的样本测试后发现原始音频信号的低频MDCT系数的小数点后第一个非零位置,在经过MP3编码器压缩后不会发生变化. 基于这一特点,他们通过改变低频区6个MDCT系数的第一个非零位置来实现秘密信息的隐写. 由于该算法修改的MDCT系数固定在前面6个系数上,而对于不同的音频信号,不同位置MDCT系数抗MP3压缩能力和对音频感知影响均不相同,因此该算法对于不同风格的音频,秘密信息检测率和算法的不可感知性方面的差别都很大. 文献[8]则采用遗传算法来自适应地选择最佳的MDCT系数来隐写. 但是该算法需要计算许多参数,计算复杂度很大,且隐写容量也不高. 同时,该算法为了正确提取已隐写的秘密信息,需要记录被选择的MDCT系数的位置. 同样在MDCT系数上隐藏信息的算法还有吴国明等人[9]提出基于局部区域信噪比的自适应隐写算法,通过信噪比指标控制隐写强度,计算音频每帧的特征矢量来选择隐写的区域,在这些区域上修改MDCT系数,从而完成隐写. 文献[10]根据相邻MDCT系数能量的大小关系实现了秘密信息的隐写,该算法避免了音频帧与帧之间量化步长误差的传递,有很好的隐蔽性. 由于大部分音频编码标准都采用了子带变换和MDCT变换,因此可充分利用这些变换的低频系数鲁棒性强的特点,来抵抗量化造成的秘密信息的丢失,这是前置式隐写方案的基本出发点. 但低频系数的修改很容易造成音频感知质量的下降,从而影响隐写算法的不可感知性. 另一方面对于已压缩的音频信号,需要先进行解码,隐写操作完成后还需要再次编码,计算量较大,无法满足许多实时应用系统的要求. 从目前的研究现状来看,还没有前置式隐写算法能保证百分之百的正确提取率,也即部分秘密信息在提取之前就已经因为压缩编码丢失,从这一点来讲,这类算法更适合于音频作品的版权保护.

(3) 在MPEG压缩比特流中实现秘密数据的隐写. 音频压缩比特流中主要包含用于解码的边信息、比例因子和哈夫曼码字,这些对象均可作为隐写的对象. 文献[11]通过研究发现小幅度地调整比例因子级别不易被人耳感知,因此可通过增大或减小比例因子实现秘密信息的隐写. 文献[12]则将若干比例因子组合并分成不同的模式,根据不同的模式在比例因子中隐藏秘密信息,该算法能抵抗压缩攻击,但隐藏容量非常有限. Neubauer和Herre在音频工程协会(AES)第108次和第109次会议上提出了压缩域音频信息隐藏的框架[13,14],并提出了针对AAC音频作品的隐藏算法,但该算法在提取时需要大量的附加信息,因此其实用性值得商榷. MP3压缩比特流中近90%以上的内容为哈夫曼码字,因此哈夫曼码字是理想的隐写对象. 但MPEG标准对比特流的格式有严格的限制,随意修改码字的内容或长度极有可能导致比特流结构的混乱,导致提取端无法正常解码,因此针对哈夫曼码字的隐写算法的设计具有很大的难度. 文献[15]提出了哈夫曼码字替换的隐写算法,可直接在MP3数据流上实现秘密信息的隐藏和提取,但测试结果也表明隐写前后码字的统计分布会发生明显改变,必须进行有效的统计补偿才能保证算法的安全性.

本文针对MP3标准中哈夫曼码字的特点,提出了一种利用码字替换实现秘密信息隐写的方法. 该方法首先对MP3码表中的码字进行分类,然后对不同类的码字进行多进制方式隐写. 由于不需要作深度解码,因此算法复杂度较低,可实现实时隐写与提取. 实验结果表明,本算法可在保证不可感知性的前提下,获得较高的隐写容量和效率.

2 MP3码流结构及码字分类

2.1 码流结构

MP3是以帧为单位进行编码,每一帧又分两个颗粒,每一个颗粒包含576个频域系数. 这些系数按频率从低到高,分为大值区(big-value)、小值区(count1)和全零区(rzero),如图1(a)所示. 大值

区的系数值较大，每两个系数 $\{x,y\}$ 用一个哈夫曼码字来表示，其码流结构如图 1(b)所示，其中 Hcode 表示两个系数所对应的码字，Sign 表示这两个系数的符号，分别用一位比特来表示，如果系数值为零，则不指定符号位，当系数值超过 15 时，其超出的值则用 linbits 表示. 小值区的系数均为 −1、0 或 1，每 4 个系数 $\{v,w,x,y\}$ 用一个哈夫曼码字来表示，小值区的码流结构如图 1(c)所示. 小值区与大值区的结构基本相同，只是小值区没有 linbits 位. 全零区系数都为 0，不需要编码.

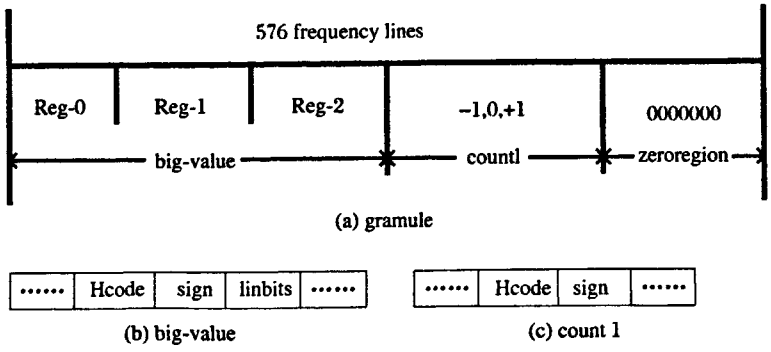

图 1　MP3 哈夫曼码流结构

Fig. 1　Structure of Huffman code stream in MP3 standard

2.2　码字分类

MP3 编码标准对比特流的格式有严格的限制，随意修改码字的内容或长度极有可能导致比特流结构的混乱，最终导致解码端无法正常解码. 因此若是需要通过修改码字实现秘密信息的隐写，就必须保证修改后的码字必须与原始码字具有相同的码字长度、符号位.（若原始码字有 linbits 位，则修改后的码字也必须保留相应的 linbits 位）. 因此，有必要在进行秘密信息的隐写操作之前，先要对 MP3 标准规定的 34 个码表中的码字进行有效分类. 对于一个给定的哈夫曼码表 T，码字分类的具体步骤如下.

(1) 把码表 T 中码字分成两类. 一类为可以被隐写的码字 C_V，即可在本码表中找到与该码字长度、符号位都相等的码字（同一码表中所有码字所对应的 linbits 位都相同）. 其余的码字则为不能被隐写的码字 C_N.

(2) 在 C_V 类中，将码字长度以及符号位都相等的码字均归为一类. 假设码表 T 共可分为 M 类码字，且每一类所含码字个数分别为 $N_1,N_2,\cdots,N_M$，要求 $N_i \geqslant 2$，记为 $\{c_{10},c_{11},\cdots,c_{1N_1}\}$，$\{c_{20},c_{21},\cdots,c_{2N_2}\},\cdots,\{c_{M0},c_{M1},\cdots,c_{MN_M}\}$，记每一类的码长和符号位分别为 $\{l_1,s_1\}$，$\{l_2,s_2\},\cdots,\{l_M,s_M\}$.

(3)及为了使隐写操作对频率系数的修改幅度降至最小，对每一类码字，按频域系数值从小到大做字典排序，以保证相邻两个码字的系数值最接近，排序之后，给每个码字标上序号，如码字 c_{in} 表示第 i 类中第 n 个码字.

表 1 给出了 MP3 标准中第 9 个哈夫曼码表经上述规则分类后，码长和符号位分别为 $\{l_i=7,s_i=2\}$，$\{l_j=8,s_j=2\}$ 的两类码字（其余类的码字表中未给出）. 从表 1 中可以看到，相邻码字的系数值都比较接近.

表 1　分类后码表 9 中的两类码字

Tab. 1　Codewords in the 9th Huffman table after Classification

$l_i=7,s_i=2$						$l_i=8,s_i=2$					
码字	x	y	码字	x	y	码字	x	y	码字	x	y
c_{i0}	2	4	c_{i3}	4	3	c_{j0}	1	5	c_{j3}	4	4
c_{i1}	3	4	c_{i4}	4	2	c_{j1}	2	5	c_{j4}	5	3
c_{i2}	3	4	c_{i5}	5	1	c_{j2}	3	5	c_{j5}	5	2

3 算法原理

3.1 混合进制密写

码字经过上述分类之后,在同一类码字中,可以通过相邻两个码字(c_{in-1},c_{in})互相替换来实现隐写,替换规则如下:假如待隐写秘密信息 $w_r=1$,原始码字为 c_{in},当 n 为偶数时,则用相邻码字 c_{in-1} 来替换原始码字. 当 n 为奇数时,则原始码字保持不变. 同理,假如待隐写信息 $w_r=0$,当原始码字 n 为偶数时,则不进行替换,否则用码字 c_{in-1} 来替换. 提取秘密信息时,只要判断含密码字所含 n 的奇偶性,就可获得秘密信息.

上述隐写过程属于二进制隐写,即每个码字最多只能隐写1比特. 但是从码字分类结果可以看到,在许多类中,相邻多个码字(一般为4个码字)其幅值都比较接近,因此可以通过多码字之间互相替换来实现多个秘密比特的隐写. 在常用的十进制或二进制中,每一位的基都是相同的(10或2),事实上,也可以定义不同基的进制系统,即混合进制系统[16]. 记 $x=(d_{n-1}d_{n-2}\cdots d_1d_0)_{b_{n-1}b_{n-2}\cdots b_1b_0}$,$x$ 是混合进制表示的值,d_i 是第 i 位的值,b_i 是第 i 位对应的基,x 的展开式为

$$x=d_0+\sum_{i=1}^{n-1}(d_i\prod_{j=0}^{i-1}b_j) \tag{1}$$

假设第 i 类所含码字个数为 N_i,若 N_i 大于4,则以4进制方式进行隐写,而对于 N_i 为2或3的类,就以2或3进制方式隐写. 在隐写过程中,每两个码字为一个操作单位,也即每一次隐写的秘密数据采用一个混合进制表示的两位数来表示,由式(1)可以得到该混合进制表示的两位数为

$$w_R=d_1+d_0\times(b_1)^1\leqslant b_0\times b_1 \tag{2}$$

其中 w_R 为一次隐写的数据量,b_0,b_1 分别是混合进制数的十位和个位所对应的基,也即两个码字所对应的基,而 d_0,d_1 分别是十位和个位的数值,也即这两个码字分别为待隐写的数据值.

3.2 算法描述

假设MP3所有的码表都已经过分类,则隐写方法的具体步骤如下.

(1) 对MP3文件以帧为单位进行解码,如果解码得到的码字属于 C_N 类码字则跳过,继续解下一个码字,直到两个码字都为 C_V 类码字. 假设解码得到两个码字分别为 $\{c_{in_0},c_{jn_1}\}$,先通过码字所属类中所含码字个数,确定各自所对应的基 $\{b_0,b_1\}$. 由 b_0,b_1 确定当前可隐写的位数 R 为

$$R=\lfloor\log_2^{b_0\times b_1}\rfloor \tag{3}$$

其中 $\lfloor\cdot\rfloor$ 表示向下取整. 从待隐写0/1序列中取 R 个秘密信息比特,并转化成十进制值 w_R.

(2) 将 w_R,b_0,b_1 代入式(2),分别求出两个码字最终所需要隐藏的信息 $\{d_0,d_1\}$ 为

$$d_1=w_R \bmod b_1 \tag{4}$$

$$d_0=((w_R-d_1)/b_1)\bmod b_0 \tag{5}$$

(3) 通过替换将 $\{d_0,d_1\}$ 隐藏到当前码字对 $\{c_{in_0},c_{jn_1}\}$. 对于第一个码字 c_{in_0},有

$$c_{in'_0}=\begin{cases}c_{in_0}, & \text{if } n_0 \bmod b_0=d_0\\ c_{i(n_0-v)}, & \text{otherwise}\end{cases} \tag{6}$$

其中,$v=(n_0 \bmod b_0)-d_0$. 当 $n'_0\geqslant N_i$ 时,需要把 n'_0 调整为:$n'_0=n_0-v-b_0$. 对于第二个码字 c_{jn_1},可同样按上述步骤完成数据隐写,得到含密码字 $c_{jn'_1}$.

(4) 重复步骤(1)~(3),直到秘密数据全部隐写完成.

在获得含密MP3音频之后,可通过以下步骤完成秘密信息的提取:首先将含密码字的序号 $\{n'_0,n'_1\}$ 分别对其基 $\{b_0,b_1\}$ 取模得到 $\{d_0,d_1\}$,然后利用式(4)和式(5),即可得到秘密信息 w_R,再将其转换成二进制比特. 待所有码字全部提取完后,就得到最终的秘密信息.

下面以一个实例来描述上述隐写和提取过程. 假设当前码字对为 $\{c_{i5},c_{j2}\}$,且各自所属类中的码字个数分别为 $\{N_i=7,N_j=3\}$,由3.1可知它们对应的基为 $\{b_0=4,b_1=3\}$. 由式(3)可得到当前可隐写的比特数 $R=\lfloor\log_2^{4\times3}\rfloor=3$. 从待隐藏的0/1序列中读取3位秘密信息,假设为110,并将其转换成十进制值 $w_R=7$,然后由式(4)和式(5)可知,码字对真正需要隐写的信息为 $\{d_0=2,d_1=1\}$. 本例中,原始码字的序号分别为 $\{n_0=5,n_1=2\}$,由式(6)可知,$n'_0=5-(5 \bmod 4)+2=6$,$n'_1=1$,也即替换后的码字对为 $\{c_{i6},c_{j1}\}$.

在提取时,只要对含密码字对 $\{c_{i6},c_{j1}\}$ 的序号及对应的基分别取模,即可得到隐写的信息 $\{d_0=2,d_1=1\}$,然后通过式(4),可得到 $w_R=7$,最后将其转换成二进制,获得原始的秘密信息110.

4 实验结果及分析

一般来讲,隐写速率定义为bit/s,考虑到MP3编码器以颗粒(granule)为单位进行编解码,而颗粒与时间之间存在转换关系,即1granule =

$F_s/576$ s，因此本文采用比特/颗粒来表示隐写速率，即 $ER = m/g$ (bit/granule)，其中 m 表示最终隐藏的总比特数，g 为原始 MP3 音频的总颗粒数. 同时，从隐写过程可知，每完成一次隐写操作时，当前码字可能会被替换，也有可能不变. 使用不同的隐写方式(二进制或混合进制)，当隐藏相同的秘密数据时，被替换的码字数是不相同的，本文用 $\eta = m/r$ 来表示隐写效率，其中 r 为被替换的码字个数. 对于二进制方式，由于秘密信息为 0/1 随机序列，可知其隐写效率 $\eta = 2$，即隐藏 m 个秘密比特，约有 $m/2$ 个码字需要进行替换. 而对于混合进制方式，其隐写效率与实际码流中码字所对应基的分布有关.

表 2 给出了 5 种不同风格的音频载体，分别在二进制和混合进制方式下的隐写容量及隐写效率(128 kbps，44.1 KHz，10 s)，可以发现混合进制方式下，无论是隐写容量，还是隐写效率都有所提高. 图 2 为同一个 MP3 音频载体(country. mp3)，在 5 种压缩比特率下使用混合进制密写时的隐写速率及效率. 从图 2 中可知，随着压缩比特率增大，隐写容量及效率都有不同程度的提高.

表 2　最大隐写速率及效率

Tab. 2　Maximum embedding ratio and efficiency on various cover audios

音频载体	颗粒数	最大隐写容量 (b/g)		隐写效率	
		二进制	混合进制	二进制	混合进制
country	1512	96.26	132.37	2.00	2.34
folk	1512	119.93	159.50	2.01	2.32
classical	1512	92.55	117.54	2.00	2.25
blues	1512	103.39	122.00	2.01	2.18
pop	1512	111.51	152.77	1.99	2.34

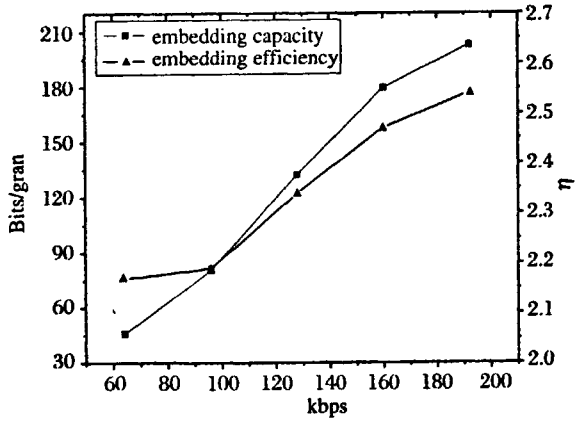

图 2　不同比特率的隐写速率及效率

Fig. 2　Embedding ratio and efficiency on various compression ratio

本文采用客观区分度 ODG 指标[17]来衡量秘密信息隐写对载体造成的失真，其中参考音频为原始音频经压缩解压后得到的时域 WAV 音频，测试音频为原始音频经隐写(含压缩)再解压得到的时域 WAV 音频. ODG 的数值范围一般在 −5～0，其值越接近于 0 表示测试音频与参考音频的相似度越高. 图 3 为不同载体在不同比特率下使用混合进制方式隐写时的不可感知性测试结果，从图 3 中可以看到，不同载体测试得到的 ODG 值均在 −1～0 范围内，而且随着比特率的增加感知质量就越好. 图 4 为分别使用二进制和混合进制方式，在相同秘密信息量的条件下不可感知性测试结果. 横坐标为隐写速率(比特/颗粒)，从图 4 中同样可以看出，混合进制方式下，算法的不可感知性要优于二进制方式.

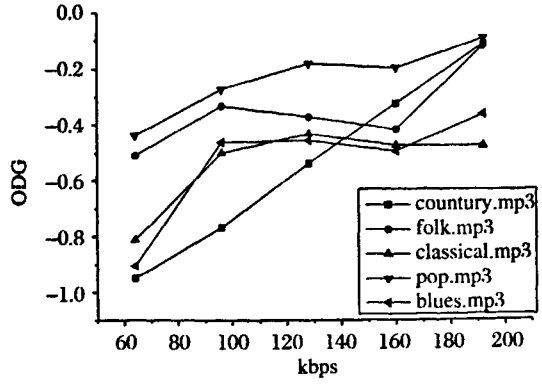

图 3　比特率与感知度的关系

Fig. 3　Relationship between compression ratio and perceptual quality

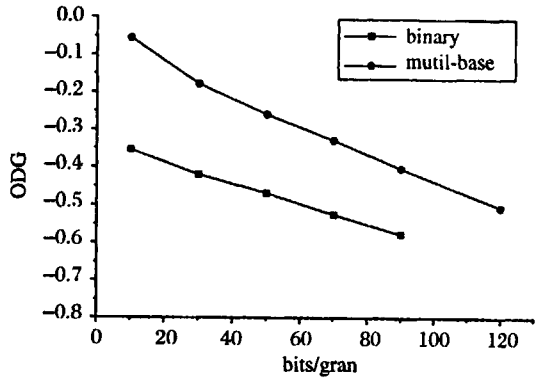

图4 相同隐写速率下两种模式的感知质量曲线

Fig. 4 Perceptual quality in binary mode and multi-base mode

5 结 语

本文利用MP3编码标准中哈夫曼编码特点，提出了一种大容量隐写算法.借助对码字进行的有效分类，避免了由于隐写而造成码流结构混乱的问题，整个隐写过程在码字层完成，不涉及深度解码和二次编码，可实现实时隐藏和提取.同时比较了二进制和混合进制两种不同模式下隐写的实验结果，表明所提出算法在隐写速率、隐写效率及保持音频感知质量方面都有一定的优势.

参考文献：

[1] Bender W, Gruhl D, Morimoto N, *et al*. Techniques for data hiding[J]. IBM System Journal, 1996, 35 (3-4): 313.

[2] Gruhl D, Lu A, Bender W. Echo hiding[C]//International Workshop on Information Hiding. London: Springer, 1996.

[3] Chen O T, Wu W C. Highly robust secure and perceptual-quality echo hiding scheme[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2008, 16(3): 629.

[4] Petitcolas F. MP3Stego[EB/OL]. (1998-12-26). [2010-03-24] . http://www. petitcolas. net/fabien/steganography/mp3stego/.

[5] Westfeld A. Detecting the low embedding rates[C]. London: Springer, 2003.

[6] 梁敬弘，王道顺，黄连生，等. 对MP3Stego的攻击研究[J]. 计算机辅助设计与图形学学报，2003，15 (8)：954.

[7] Wang C T, Chen T S, Chao W H. A new audio watermarking based on modified discrete cosine transform of MPEG/Audio Layer III[C]. Washington: IEEE, 2004.

[8] Moghadam N, Sadeghi H. Genetic content-based MP3 audio watermarking in MDCT domain [C]. Rome: WASET, 2005.

[9] Wu G M, Zhuang Y T, Wu F, *et al*. Adaptive audio watermarking based on SNR in localized regions[J]. Journal of Zhejiang University Science A, 2005, 6: 53.

[10] 刘伟，王朔中，张新鹏. 一种基于部分MP3编码原理的音频水印[J]. 中山大学学报：自然科学版，2004，43(2)：26.

[11] Qiao L, Nahrstedt F. Non-invertible watermarking methods for MPEG encoded audio[C]. Bellingham: SPIE, 1999.

[12] Koukopoulas D K, Stamatiou Y C. A compressed-domain watermarking algorithm for MPEG audio layer III[C]. New York: ACM, 2001.

[13] Neubauer C, Herre J. Audio watermarking of MPEG-2 AAC bitstreams[C]. Pairs: Audio Engineering Society, 2000.

[14] Neubauer C, Herre J. Advanced Watermarking and its Applications[C]. California: Audio Engineering Society, 2001.

[15] 高海英. 基于Huffman编码的MP3隐写算法[J]. 中山大学学报：自然科学版，2007，46(4)：32.

[16] Zhang X P, Wang S Z. Steganography using multiple-base notational system and human vision sensitivity[J]. IEEE Signal Processing Letters, 2005, 12 (1): 67.

[17] Thiede T, Treurniet W C, Bitto R, *et al*. PEAQ-the ITU standard for objective measurement of perceived audio quality [J]. Journal of the Audio Engineering Society, 2000, 48(1&2): 3.

[责任编辑：伍少梅]