基于 Huffman 编码的 MP3 隐写算法*

高海英
（解放军信息工程大学电子技术学院，河南 郑州 450004）

摘 要：针对 MP3 音频的编码特点，提出了基于 Huffman 码字替换原理的音频隐写算法。与以往的 MP3 隐写算法相比，该算法直接在 MP3 帧数据流中的 Huffman 码字上嵌入隐蔽信息，不需要局部解码，具有透明度高、嵌入量大、计算量小的特点。通过实验分析了算法的透明性、嵌入量、码字的统计特性等方面的特点。

关键词：MP3 编码；Huffman 编码；隐藏容量；平均信噪比
中图分类号：TP309 **文献标识码**：A **文章编号**：0529-6579（2007）04-0032-04

由于 MPEG－1 Layer III（MP3）有很高的压缩比和较小的失真率，使得 MP3 成为存储音乐、歌曲的主流格式。目前，信息隐藏领域针对 MP3 文件的研究主要分为两个方面：一是用于 MP3 文件的知识产权或版权保护的数字水印技术；二是用于以 MP3 文件为载体的隐蔽通信的隐写和隐写检测技术。

针对 MP3 的隐藏技术大致分为两类：一类是针对 WAV 等未压缩的音频文件设计隐藏算法，要求算法能够抵抗 MP3 压缩解压，即把嵌入隐蔽信息的未压缩音频文件进行 MP3 压缩后再传送给对方，对方对接收到的 MP3 文件首先进行解压，从解压后的文件中仍然能够正确提取隐蔽信息。例如，文献［1］在 2001 年提出了使用扩频技术和 HAS 掩蔽效应将水印嵌入到非压缩的原始音频时域信号中的盲水印算法；另外一类是针对 MP3 编码的特点，在 PCM 流压缩过程中嵌入隐蔽信息，或者对 MP3 文件进行局部解码，找到嵌入位置，嵌入隐蔽信息后再进行编码。例如，常用的音频隐藏工具 MP3Stego 就是通过修改量化和编码后的长度作为信息隐藏的方法[2]。文献［3］则通过修改 MDCT 变换后的低频系数嵌入隐蔽信息。文献［4］是在 MP3 码流中嵌入数据，将隐蔽信息加载在尺度因子域中（scale－factor domain），而不是加载在基于掩蔽效应的数字音频数据或系数上，嵌入时并不需要进行 MP3 编码。

上述两类技术存在一定的弊端，首先第一类在嵌入隐蔽信息后需专门经过 MP3 压缩得到 MP3 格式的文件，而接收方必须经过解码后才能提取信息，第二类是嵌入隐蔽信息的过程需要进行解码和编码，提取隐蔽信息的过程需要进行局部解码，这两类方法都不是直接对 MP3 文件进行操作，在嵌入和提取过程中计算量比较大，因此需要找到一种方法具备以下特点：直接对 MP3 帧数据流进行操作，不需要局部解码和编码，透明性高，容量大的隐藏算法。隐藏算法的用途不同，对算法的设计要求也就不同，用于版权保护的 MP3 数字水印算法对鲁棒性要求较高，透明性和容量次之，而本文研究的是用于隐蔽通信的 MP3 隐写算法，隐写算法对透明性和容量要求较高，鲁棒性次之[5－10]。

文章通过修改 MP3 帧数据中的 Huffman 码字设计了一种 MP3 隐写算法，该算法不需要对 MP3 文件进行局部解码，具有隐藏速度快、透明性高、嵌入量大的特点。并且通过实验分析了算法的透明性、嵌入量、码字的统计特性等方面的特点。

1 MP3 编码

MP3 编码算法的过程如图 1 所示。

MP3 编码算法流程分为三部分：时频映射、心理声学模型和量化与编码。其中时频映射部分包括子带滤波器组和 MDCT（修正的离散余弦变换），量化编码包括比特和比例因子分配和 Huffman 编码。

输入 PCM 音频数据是按帧进行处理的，每帧包括 1 152 个 PCM 样值，而每帧又分为两节数据，也就是每节数据包含 576 个 PCM 样值。MP3 的压缩算法实质上属于有损压缩，而对于人耳来说，MP3 的压缩算法属于无损压缩。这里应用的理论

* 收稿日期：2006－09－04
基金项目：国家自然科学基金资助项目（90604022）
作者简介：高海英（1978 年生），女，讲师，博士；E-mail：coconutghy@tom.com

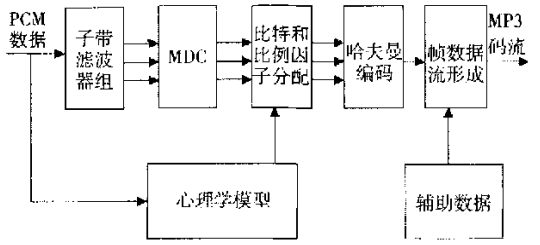


图 1　MP3 编码算法结构框图

Fig. 1　Sketch of the basic structure of the MP3 encoder

基础是人耳的听觉系统的掩蔽效应，包括时域掩蔽和频域掩蔽效应，主要是应用频域掩蔽效应。为了应用频域掩蔽效应，需对每节的 576 个 PCM 样值作时频变换，首先将 PCM 样值送入子带滤波器值，经子带滤波器组均匀地分为 32 个子带信号，每个子带包含 18 个样值。然后，再对各子带作 MDCT 变换，从而得到 576 个等间隔的频域样值。

经时频变换后得到的左右声道频域样值需根据所要求的模式进行声道模式处理，频域样值经模式处理后，就进行量化。编码算法采用的是心理声学模型控制下的非均匀量化，将量化的结果送至 Huffman 编码器进行无损编码，这样会增加算法的复杂度，但可以利用信号的统计特性提高压缩率。这也是 MP3 压缩算法与 MPEG－1 的层一和层二的主要区别之一。

编码算法的最后一步是比特流打包形成的 MP3 码流。也就是按照 MP3 标准所规定的码流格式，把帧头、纠错码、副信息、主数据、附加数据等有关信息组合成适合于解码的帧。因此，这个过程也称为帧生成过程。

2　MP3 隐写算法

MP3 中每帧数据的主数据包含了每个比例因子频段的比例因子值和 Huffman 编码数据，该算法通过改变部分 Huffman 码字，达到嵌入隐蔽消息的目的，而且在听觉上与原始音频无任何差异。

2.1　嵌入算法

搜索载体 MP3 文件的主数据中的特定的 Huffman 码字 HCode，例如码长是 6 的一对码字 0x18 和 0x1b，假设在码字 0x18 上嵌入隐蔽消息 Ciper = 0，在码字 0x1b 上嵌入隐蔽消息 Ciper = 1，嵌入隐蔽消息后的码字为 HCode′，令码长为 CodeLen，则具体嵌入方法如式（1）：

$$\mathrm{HCode}' = \begin{cases} 0x18, \text{if Codelen} = 6, \text{Hcode} = 0x18, W = 0; \\ 0x1b, \text{if Codelen} = 6, \text{Hcode} = 0x18, W = 1; \\ 0x18, \text{if Codelen} = 6, \text{Hcode} = 0x1b, W = 0; \\ 0x1b, \text{if Codelen} = 6, \text{Hcode} = 0x1b, W = 1. \end{cases} \quad (1)$$

用 HCode′代替 HCode，重新对比特流进行打包形成携密 MP3 码流。

2.2　提取算法

搜索携密 MP3 文件的主数据中码长是 6 的 Huffman 码字 0x18，0x1b。具体提取方法如式（2）：

$$W = \begin{cases} 1, if\ CodeLen = 6, Hcode = 0x1b; \\ 0, if\ CodeLen = 6, Hcode = 0x18 \end{cases} \quad (2)$$

3　Huffman 码对的选取

该隐写算法的基本思想是对一些特定的 Huffman 码字进行修改，从而达到嵌入隐蔽消息的目的。下面详细介绍选取 Huffman 码对的方法。

（1）用于隐写的码对是等长的，例如，原码字是 6 比特长，修改后的码字也必须是 6 比特长。

（2）MP3 标准中共有 32 个码表，详见 iso11172_ 3。用于隐写的码字必须是成对选取，并且保证码对出现在同一个码表中。

（3）不要选取码字对应的 x 或 y 中有 0 出现的码字，若原码字对应的 x 是 0，替换后的码字对应的 x 是 1，则解码过程中就多读取一个符号位，这样造成了以后的比特流的错位，会造成解码错误。

（4）选取的码字对应的 x、y 应很接近，例如 8 长码字 0x3d 对应的 $x=1$，$y=6$，8 长码字 0x33 对应的 $x=1$，$y=7$，8 长码字 0x2a 对应的 $x=2$，$y=8$，在这种情况下，选取 0x3d 和 0x33 作为码对，而不选取 0x2a 和 0x3d，这样做尽量减少数据上的差异。

该算法选取的码字有以下 13 对，如表 1 所示。

满足上述要求的码对有很多，若需要增加算法的嵌入量，可以选取更多的码对。

4　隐藏算法的分析

4.1　容量分析

该隐写算法的原理是在特定的 Huffman 码字上嵌入隐蔽信息，而对于不同的 MP3 文件，这些特定码字的个数也不同，因此在嵌入隐蔽信息之前，需要判断载体文件的隐藏容量。载体 MP3 文件的隐藏容量的计算等价于求 MP3 文件中上述 13 对码

表1 Huffman码对

Tab. 1 Huffman codes

x	y	hlen	Hcode
2	3	6	011000 (0x18)
1	3	6	011011 (0x1b)
3	1	6	011100 (0x1c)
3	2	6	011001 (0x19)
2	4	7	0101001 (0x29)
2	5	7	0100010 (0x22)
5	1	7	0100101 (0x25)
5	2	7	0100011 (0x23)
1	4	7	1000111 (0x47)
2	4	7	1000100 (0x44)
1	6	8	00111101 (0x3d)
1	7	8	00110011 (0x33)
7	1	8	00110101 (0x35)
7	2	8	00110001 (0x31)
4	3	8	10000111 (0x87)
4	4	8	01111111 (0x7f)
3	7	9	001011101 (0x5d)
4	7	9	001001111 (0x4f)
10	7	10	0000111001 (0x39)
10	8	10	0000110000 (0x30)
5	6	11	00010101101 (0xad)
5	5	11	00010111001 (0xb9)
8	5	11	00010001111 (0x8f)
8	4	11	00010011100 (0x9c)
5	10	12	000011101000 (0xe8)
5	9	12	000011111101 (0xfd)

字的个数，码字个数决定了能嵌入的隐蔽消息的最大比特数。若隐蔽消息的比特数超过了最大容量，需另做处理。表2是试验MP3文件的容量，从表2可以看出，载体大小差异不大的情况下，容量存在很大差异。

表2 隐藏容量

Tab. 2 The capacity of hiding

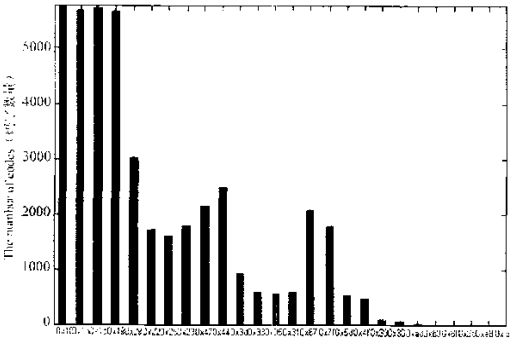
载体MP3文件/kb	容量(byte)
3.224	13 339
2.885	8 064
2.683	5 563
4.598	15 394
2.952	6 470

4.2 算法透明性分析

从量化的角度衡量隐藏前后音质的差别主要采用的性能指标是分段平均信噪比SNR[5]。通过实验求10^5个帧（每帧256个采样数据）的携密语音的平均信噪比（dB）≈67.48，人耳很难察觉到这个差别。

4.3 码字统计量分析

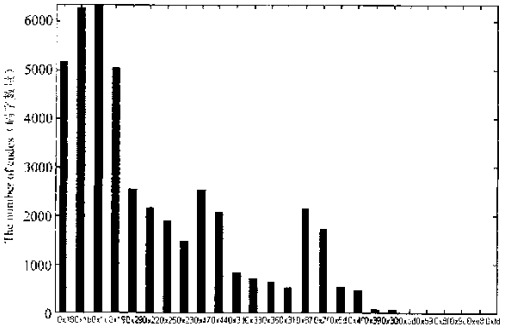
从算法的嵌入原理可以看出，算法的鲁棒性较差，修改前后的码字的统计分布必然会发生一些变化。嵌入隐蔽信息前后使用的码字统计量的变化如图2和图3所示。



codes（码字）

图2 载体码字的分布

Fig. 2 The distribution of original file's codes



codes（码字）

图3 携密文件码字的分布

Fig. 3 The distribution of hidden file's codes

使用的码字的统计分布的改变是该算法的一个缺陷，但需要说明的是，在iso11172－3标准中，符合3中要求的码对有很多，隐藏检测者在不知道算法使用的具体码对的情况下，很难从统计规律上判断MP3文件中是否携带了隐蔽信息，因此算法中使用的码对必须是保密的。

5 结 语

文章基于MP3数据流的Huffman码字的替换原理设计了一种MP3隐写算法，该算法具有听觉效果好、隐藏容量大、计算量小的特点，由于MP3文件在网络中的广泛传播，该隐写算法具有

很大的实用性。当然，该算法需要进一步改进，这些特殊码字经过改写后，必然引起码字统计分布的改变，怎样进行统计补偿，是作者有待考虑的问题。

参考文献：

[1] CVEJIC N, KESKINARKAUS A, SEPPANEN T. Audio watermarking using m－sequences and temporal masking. Applications of Signal Processing to Audio and Acoustics [C]. 2001 IEEE Workshop on the 21－24 Oct. 2001: 227－230.

[2] PETITCLOLAS F A P. http://www.petitolas.net/fabien/steganography/mp3stego/index.html. 2006.06.

[3] WANG Ching－Te, CHEN Tung－Shou, CHAO Wen－Hung. A new audio watermarking based on modified discrete cosine transform of MPEG/audio layer III[C]. Networking, Sensing and Control, 2004 IEEE International Conference on Volume 2, 2004: 984－989.

[4] NEUBAUER C, HERRE J. Advanced audio watermarking and its applications[C]. In 109th AES Conv., USA, 2000, Preprint 5176.

[5] 赵春晖，李福昌. 数字音频水印技术：回溯与展望[J]. 哈尔滨工程大学学报，2002，23(6)：57－61.

[6] XU C, WU J, SUN Q. Digital Audio Watermarking And its Application in Multimedia Database[C]. In 5th Int. Symposium on Signal Processing and its Application, Australia, 1999: 91－94.

[7] BASSIA P, PITAS I, NIKOLAIDIS N. Robust audio Watermarking in the time domain[J]. IEEE Transactions on Multi media, 2001, 3 (2): 232－241.

[8] NEUBAUER C, HERRE J. Audio watermarking of MPEG－2 AAC bitstreams[C]. In 108th AES Conv. Rance, 2000, Preprint 5101.

[9] PETER NOLL. MPEG audio digital coding[C]. IEEE Signal Processing Magazine, 1997: 1053－5888.

[10] PODILCHUK C I, DELP E F. Digital watermarking: algorithms and applications[C]. IEEE Signal Processing Magzaine, 2001: 33－46.

The MP3 Steganography Algorithm Based on Huffman Coding

GAO Hai-ying

(Electronic Technology Institute, PLA Information Engineering University, Zhengzhou 450004, China)

Abstract: A MP3 Steganography algorithm is introduced based on replacing Huffman codes with the characters of MP3 coding. Compared with ancient steganography algorithms, the algorithm directly embeds secret information in Huffman codes of MP3 frames, and it need not partly decode. It has the characters of transparency, big capacity and low computing complexity. Experiments analyze the characters of the algorithm.

Key words: MP3 coding; steganography algorithm; Huffman coding; average signal noise ratio