一种基于改进的分段 DCT 域多格式音频水印算法*

施荣华，潘　登，刘卫国，吴　倪
（中南大学 信息科学与工程学院，长沙 410075）

摘　要：针对因为格式种类庞大而导致音频文件数字水印的嵌入较复杂且繁琐的问题，采取了利用 FFMPEG 将音频文件转换成一种统一格式而进行数字水印嵌入的方法，其效率和效果有较大提高。对于水印算法提出了一种改进的分段 DCT 算法，将二维的水印图像转换成一维数字序列并进行置乱加密和极化操作，然后对数字音频信号作分段 DCT 变换，根据每段的能量不同采取不同等级的嵌入强度，其中选取四个不同等次的系数进行反复嵌入。实验仿真表明，在经过格式转换后的音频文件与原始文件之间没有出现失真，该算法在经过加入高斯白噪声、低通滤波、有损压缩等操作后仍具有很强的稳健性。

关键词：音频；多格式；嵌入强度；分段；离散余弦变换
中图分类号：TP302　　**文献标志码**：A　　**文章编号**：1001-3695(2013)12-3779-04
doi:10.3969/j.issn.1001-3695.2013.12.067

Multi-format audio watermarking algorithm based on improved quantizing DCT domain

SHI Rong-hua, PAN Deng, LIU Wei-guo, WU Ni
(*School of Information Science & Engineering, Central South University, Changsha* 410075, *China*)

Abstract: According to the various formats cause the phenomenon of audio file's watermarking embedding becomes complex and complicated, this paper used FFMPEG to make the various formats transform into a unity format to embed the watermarking which improved the efficiency and result. it selected improved quantizing discrete cosine transform algorithm to embed the watermarking. First, it converted the binary image into the one dimensional sequence and modulated the sequence with scrambling encryption and polarization operation, then performed the digital audio signal the quantizing DCT transform, according to each different subparagraphs energy grade, it used different embedding strength and selected four different levels coefficients, repeated embedding. In the experimental simulation results, there doesn't exist obvious differences between the transformed audio signal and the original audio signal, the watermark algorithm is imperceptible to digital audio signal such as Gauss white noise interfere, low pass filter and loss compression.

Key words: audio; multi format; embedding strength; quantization; discrete cosine transform

0　引言

互联网的蓬勃发展使得数据的交换和传输变得快捷，尤其是多媒体数据中的音频文件在互联网中占有很重要的地位。因为它在互联网中能轻易获取，随之而带来的副作用也是非常明显的[1]。通过 Internet 肆无忌惮地复制和传播盗版音乐制品，通过网络传输数据文件或作品使有恶意的个人或团体有可能在没有得到作品所有者的许可下复制和传播有版权的内容[2]。虽然这些问题能够通过数字水印、隐蔽信道[3]、伪装式保密信道[4]等信息隐藏的方法来解决，但是实际解决水印嵌入时对于各种格式的音频文件，这些对策往往也是一筹莫展。因为不同格式的音频文件的编码不同，对于实际的数字水印嵌入系统来说，所采用的嵌入方法也不同，每种格式需对应一种方法。对此，不仅方法的实现有困难，而且实现后的系统繁琐度太高。

针对音频文件格式的复杂性并利用实际工具解决这一问题的研究较少，而基于 DCT 域[5~7]的水印嵌入算法主要是将嵌入点集中在能量最高处[8,9]，结合 AC 和 DC 分量的水印嵌入[10]及信噪比的自适应嵌入[11]。本文首先采用 FFMPEG 源程序实现了音频格式的转换问题，实现了格式的统一化，水印嵌入采用的是基于分段 DCT 域能量等级的嵌入对策，对于各种音频攻击的特性并且结合分段 DCT 域能量的不同性质，进行了不同的水印强度嵌入，增强了水印的鲁棒性。

1　多格式的转换

1.1　FFMPEG

对于音频文件格式的转换采用的是开源程序 FFMPEG。FFMPEG 是一个包含录制、音频视频编/解码、格式转换等功能为一体的开源解决方案。它的开发是基于 Linux 操作系统，但

收稿日期：2013-03-09；**修回日期**：2013-04-16　　**基金项目**：国家自然科学基金资助项目(61272495,60773013)；长沙市科技计划基金资助项目(K1104009-11)

作者简介：施荣华(1963-)，男，湖南常德人，教授，博士，主要研究方向为密码学、信息安全；潘登(1989-)，男，湖南岳阳人，硕士研究生，主要研究方向为信息安全、数字水印(cl198896@hotmail.com)；刘卫国(1963-)，男，湖南祁阳人，教授，博士，主要研究方向为网络与信息安全、智能信息处理；吴倪(1987-)，女，湖南益阳人，硕士研究生，主要研究方向为信息安全.

在其他操作系统中也能兼容使用,它没有成型的 GUI,故只能在命令提示符的界面下进行操作。FFMPEG 支持 MPEG、MPEG-4、FLV、DV 等 40 多种编码以及 AVI、OGG、MPEG 等 90 多种解码。FFMPEG 能从音频源和视频源获取数据并保存,快速实现音频视频格式的转换。

1.2 格式转换

利用 FFMPEG 实现音频文件格式的转换,实现过程范例如下:

ffmpeg-i inputfile. mp3-f wma-acodec libaac-vn-ab 64k outputfile. wma

其中:i 是设定输入的文件名,如本例的 inputfile. mp3;f 是设定输出文件的格式,如本例为 WMA;acode 是设定声音的编/解码器,本例采用的是 AAC 编码;vn 是说明不对视频作处理、只针对音频作处理时使用;ab 是设定每个信道的流量,本例设定的是 64k;outputfile. wma 是输出音频的名字以及格式。

FFMPEG 还可以实现其他格式的音频文件的转换,只需在相应的参数进行对应的调整。

1.3 格式转换前后对比

比较转换前后波形图的测试软件采用的是 Cool Edit。首先考虑 MP3 以及 WAV 格式转换为 WMA 格式的情况。MP3 是有损压缩编码格式,而标准格式的 WAV 文件与 CD 格式一样,都是 44.1K 的采样频率,速率为 88 kps,且 16 位的量化位数。WMA 在比特率低于 128 kbps 时,在所有音频格式中的表现是最好的;但当比特率高于 128 kbps 时,WMA 在音质方面不会有明显的改变。MP3 却不一样,当比特率高于 192 kbps 时,MP3 所表现的效果会有明显提高。考虑到转换后的文件大小问题,WMA 格式在所有音频文件中尺寸是相对较小的,符合实际的需要。

实验中分别选取 MP3 和 WAV 格式的音频文件为源文件,采用 FFMPEG 源程序进行格式转换,在代码中对转换后的参数进行相应的调节,可得到两种格式分别对应的 WMA 音频文件。

在 Cool Edit 中,本文分别对转换前后的音频文件进行采样、编辑,观察两者的波形图。图 1 为转换前 MP3 格式的音频源信号,单声道,比特率为 320 kbps,8 bit 量化。图 2 为经过 FFMPEG 转换后格式为 WMA 的音频信号。

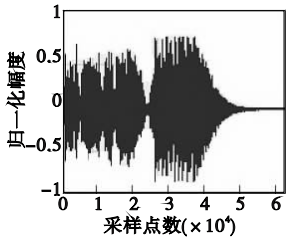

图1 转换前MP3格式的音频源信号

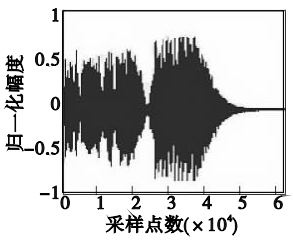

图2 经过FFMPEG转换后格式为WMA的音频信号

根据两幅波形图的效果,并没有出现波形不和谐的情况,波峰之间的差距不大,没有造成明显的失真,在听觉效果方面与原始音频信号听起来没有差别。

图 3 为转换前 WAV 格式的音频源信号,单声道,16 bit 量化。同理,根据以上观察,WAV 格式的音频文件的转换也是相当成功的,如图 4 所示。

对于其他不常用的格式,如 OGG 格式的音频文件,它是一种新的音频压缩格式,与 MP3 格式类似,故采用 FFMPEG 转换时,只需用户对相关参数进行设置便能转换为 WMA 格式的音频文件。对于另一种有损音频文件格式 AAC,它采用的是一种高压缩比的音频压缩算法,具有很强的复杂性,使用这一格式的音频并不多。在利用 FFMPEG 转换时,对音质会有一定的损伤,波形会产生一定的失真,但在听觉方面不会造成太大的影响,当经过水印的嵌入和提取后,转换后的音乐效果与原始音乐基本上没有差别。

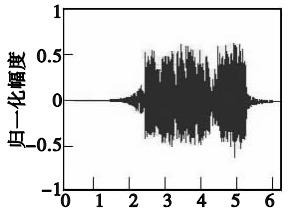

图3 转换前WAV格式的音频源信号

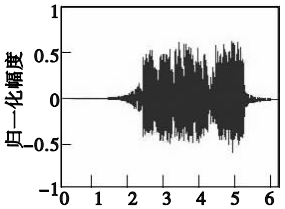

图4 经过FFMPEG转换后格式为WMA的音频信号

2 音频文件的数字水印算法

2.1 降维处理

本文所使用的水印 W 是大小为 $M_1 \times M_2$ 的二维二值图像,它可表示为

$$W = \{w(m,n), 0 \leqslant m \leqslant M_1 - 1, 0 \leqslant n \leqslant M_2 - 1\}$$
$$w(m,n) \in \{0,1\} \quad (1)$$

要将其嵌入到一维的数字音频信号 S 中,需将其进行降维处理,将二维水印图像 W 转换成一维的数字序列 V。操作方法如下:

$$V = \{v(t) = w(m,n), 0 \leqslant m \leqslant M_1 - 1, 0 \leqslant n \leqslant M_2 - 1,$$
$$t = m \times M_2 + n\} \quad (2)$$

降维操作后,W 中的像素 $w(m,n)$ 用序列 V 中的元素 $v(t)$ 表示。

2.2 置乱加密和极化操作

数字水印一个很重要的特性是保密性。为了消除序列 V 中元素间的相关性,进一步提高水印抵抗各种干扰的能力和嵌入水印的稳健性,在对二值图像采取的加密方法是在降维操作后,使用经典 Hill 算法实现一维序列的置乱。

Hill 加密算法的基本思想是将 I 个明文字母通过线性变换将它们转换为 K 个密文字母,密钥是此变换矩阵。解密只要作一次逆变换就可以,置乱后得到新的序列 W_0:

$$W_0 = \{w_0(t), 0 \leqslant t \leqslant M_1 \times M_2\} \quad (3)$$

为增强嵌入水印的鲁棒性,对得到的新序列 W_0 进行极化处理。为了降低相邻项之间的连续性,对极化处理后的序列作伪随机排列的处理:

$$W_p = \text{permute}(W_{\text{polar}}) = \{w_p(t) = v_{\text{polar}}(W_0), 0 \leqslant t\} \quad (4)$$

2.3 水印嵌入算法

DCT 是数据压缩常用的一个变换编码方法,其变换核为实数的余弦函数。DCT 是将图像分成 $N \times N$ 像素块,然后对 $N \times N$ 像素块逐一进行 DCT。

$N \times N$ 像素矩阵 $I(i,j)$ 的 DCT 定义为

$$f(u,v) = \frac{1}{\sqrt{2N}} c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j) \times$$

$$\cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right] \tag{5}$$

其中:$i,j=0,1,\cdots,N-1$;$c(x)=\begin{cases}\frac{1}{\sqrt{2}} & \text{当 } x=0\\ 1 & \text{其他}\end{cases}$。

水印嵌入的方法是每隔一段音频文件嵌入一个对应的水印,所以首先需将原始数字音频信号进行分段。假设原始音频长度为 L,则需满足 $L\geqslant M_1\times M_2\times N\times(2^k-1)$。

对分段后的原始音频文件取其中的奇数段,假设 $e(i)$ 为原始音频,则每段的能量可用式(6)表示:

$$E(2k-1)=\sum_i e(i)^2 \tag{6}$$

将计算出的能量值按从大到小排序,并取其中的最大值 $E_{\max}$和最小值 $E_{\min}$,计算最大值与最小值之差,根据能量差值将其平均分为四份,同时意味着将所有音频段分为了四个区间。确定每个区间的嵌入强度及强度增加的比值 p_0,如第一个区间的嵌入强度为最低嵌入强度 p,则第二个区间的嵌入强度为 $p+p_0$。依此类推,后面两个区间的嵌入强度分别为 $p+2p_0$、$p+3p_0$。每个音频段都按照此规律对嵌入强度进行确定。

在不同攻击的影响下,不同的 DCT 系数对于攻击的稳健性各不相同。为了强化水印算法的鲁棒性,在进行分段 DCT 之后,将所有 DCT 系数按从小到大排序并平均分成三个区间,选取其中的高频系数、低频系数和三个区间之间两个节点的系数,共四个不同等次的系数。对这四个位置进行水印的重复嵌入,分析提取出的水印值,选取四个位置中嵌入效果最好的位置,最终确定水印值。这样就能保证在进行不同攻击的情况下,该算法能对大多数的攻击具有很好的鲁棒性。

用 $D_e(2k-1)(m_1)$、$D_e(2k-1)(m_2)$、$D_e(2k-1)(m_3)$、$D_e(2k-1)(m_4)$分别表示上述四个系数值,对其进行参数量化过程,如图 5 所示。

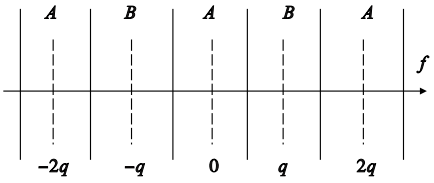

图5　双极性参数量化的量化

a)选取量化步长 q,对选取的参数值域划分为 A、B 两个区间集,q 的整数倍都位于该区间的中点。

b)位于 A 区间集里的坐标都表示比特“1”,而在 B 区间里的都表示比特“0”。

c)用量化步长 q 对待量化参数 f 进行取整数商和余数运算。

$$m=\left[\frac{f}{q}\right] \tag{7}$$

$$r=f-m\times q \tag{8}$$

式中:[ ]表示从负向取整数,整数商 m,余数 r。

根据水印比特值 $w(i,j)$ 和待量化参数 f 所处在的区间集进行量化,量化表如表 1 所示。

表 1　参数 f 的量化表

	$m=2k+1$	$m=2k$
$w(i,j)=1$	$f'=mq+q$	$f'=mq$
$w(i,j)=0$	$f'=mq$	$f'=mq+q$

其中:f' 表示经过量化后的参数 f。

量化后的水印比特信息由 f' 所在区间确定,若 f' 处在 A 区间,则表示水印比特信息“1”;若 f' 处在 B 区间,则表示水印比特信息“0”。该过程可视为对参数作适当修改和赋二值信息。

对奇数段的四个系数进行上述量化过程后,再对其进行 IDCT,最终得到嵌入水印的音频文件。

2.4　水印提取算法

水印提取过程可以看做是水印嵌入的逆过程。

首先检测嵌入水印的音频信号的长度,若长度不一致,则采取样条插值的方法进行攻击补偿,以免造成水印提取错误。

样条插值是使用一种特殊的分段多项式进行插值的过程,它是使用较低的多项式样条实现较小的插值误差,从而避免了使用高阶多项式而引起的误差,并且计算的速度也较快。

将嵌入水印的音频补充到等长时,对其进行分段并取奇数段的 DCT,系数为

$$D_e(2k-1)$$

$D_e(2k-1)(m)$ 则表示第 $2k-1$ 段的 DCT 的第 m 个系数值。计算每个奇数段的能量值,根据嵌入时的算法计算各段的嵌入强度。

选取高频系数、低频系数和二值之间的值平均分为三段且位于两个节点处的值,共四个不同等次的系数 $D_e'(2k-1)(m_1)$、$D_e'(2k-1)(m_2)$、$D_e'(2k-1)(m_3)$、$D_e'(2k-1)(m_4)$,并分别计算:

$$d=\left[\frac{D_e'(2k-1)(m)}{p'(2k-1)}\right] \tag{9}$$

其中:[ ]表示取整;$p'(2k-1)$ 表示位于 $2k-1$ 奇数段的嵌入强度,它的取值分别为 p、$p+p_0$、$p+2p_0$、$p+3p_0$。

如果 d 为奇数,则处于量化区间 B 中,水印比特值 $w'(i,j)=0$;如果 d 为偶数,位于区间 A 中,水印比特值 $w'(i,j)=1$。

根据上述的四个系数提取出四个水印信息序列,选择其中出现次数最多的水印值,并确定为提取的水印信息 W'_p。

对水印信息作伪随机序列逆排列,得到提取水印的一维序列 W_0'。

$$W_0'=\text{InversePermute}(W_p')=\{w_0'(t)=v_{\text{polar}}'(W_0'),0\leqslant t\} \tag{10}$$

由于得到的水印序列 W'_0 是经过 Hill 算法加密后生成的伪随机序列,要恢复其正常序列,必须按照对应的 Hill 算法对该序列进行解密的伪随机逆序列。

$$W''_0(t)=\{w''_0(t)=w'_0(t),0\leqslant t\leqslant(M_1\times M_2-1)\} \tag{11}$$

所使用的水印图片是视觉可辨识的二值图像,因此要对逆排序后的序列 $W''_0(t)$ 进行升维处理,得到提取的水印 W_0。

$$W_0=(w_0(m,n)=w''_0(t),0\leqslant m\leqslant M_1-1,\ 0\leqslant n\leqslant M_2-1,t=m\times M_2+n\} \tag{12}$$

为了消除观测者的经验、实验条件和设备条件等主观因素的影响,采用归一化相关系数对原始水印和提取的水印的相似性作客观评价。

$$NC(w,ws)=\frac{\sum_{i=0}^{M_1-1}\sum_{j=0}^{M_2-1}w(i,j)ws(i,j)}{\sqrt{\sum_{i=0}^{M_1-1}\sum_{j=0}^{M_2-1}w(i,j)^2}}\times\frac{1}{\sqrt{\sum_{i=0}^{M_1-1}\sum_{j=0}^{M_2-1}ws(i,j)^2}} \tag{13}$$

3 仿真测试及结果分析

仿真实验采用的是经过 FFMPEG 转换后为 8 位 128 kbps 的 WMA 格式音频信号作为原始载体语音采样信号，如图 6 所示。图 7 是含水印的数字音频信号。采用图像作为水印，如图 8 所示。从仿真实验可以看出，比较原始数字音频信号和含水印的数字音频信号的波形图，两者之间失真几乎不存在，嵌入水印后信噪比 SNR = 37.26 dB，说明嵌入水印后的音频已经具有良好的音频效果，从而证明了采用 FFMPEG 转换格式后的原始音频文件在利用本算法嵌入水印后的透明性有很好的效果。

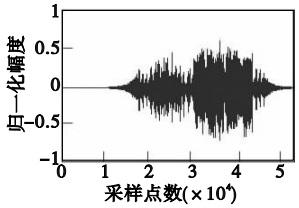

图6 原始载体语音采样信号波形图

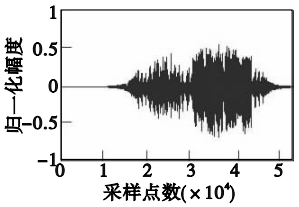

图7 嵌入水印后的数字音频波形图

为了检测经过格式转换后的音频水印文件的稳健性，对其分别进行下面的处理：a）音频压缩，压缩至 128 kbps；b）低通滤波，截止频率为 10 kHz；c）加入高斯白噪声（均值为 0，均方差为 0.01）。图 9（a）～（c）分别为经过以上处理后提取的水印，归一化相关系数分别为 0.68、0.75、0.82。从图 9 中可以得出，经过该算法处理后的提取水印与原始水印具有很强的视觉相似性，所以对数字音频信号的版权具有很好的保护性。

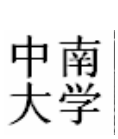

图8 插入的水印图片

图9 提取的水印图像

4 结束语

本文首先通过 FFMPEG 开源程序实现了音频文件的格式转换，比第三方软件具有更强的嵌入性和灵活性，省去了因为格式导致实际的水印嵌入复杂的繁琐过程，音频格式转换前后的失真情况几乎为零。对于本文所提出的基于改进的分段 DCT 域水印算法是根据不同的水印攻击、透明性的需要以及置乱极化操作和量化等方法和原理上研究出来的。实验结果表明，此方案在水印透明性方面有很好的成果，能抵御大多数的水印攻击，稳健性强、计算时间短、可行性很高。由于现在对音频文件的水印攻击没有一套完整的标准，文中得到的攻击数据只是在实验基础上主观得到的，而且文中提到的四个嵌入位置也不一定是水印嵌入的最佳位置，这些将是下一步研究的重点。

参考文献：

[1] 王颖，肖俊，王蕴红. 数字水印原理与技术[M]. 北京：科学出版社，2007.

[2] PODILCHUK C I, DELP E J. Digital watermarking: algorithms and applications[J]. IEEE Signal Processing Magazine, 2001, 18(4): 33-46.

[3] PAINTER T, SPANISAS A. Perceptual coding of digital audio[J]. Proc of the IEEE, 2000, 88(4): 451-515.

[4] VOYATZIS G, PITAS I. The use of watermark in the protection of digital multimedia product[J]. Proc of the IEEE, 1999, 87(7): 1197-1207.

[5] 温泉，王树勋，年桂君. DCT 域音频水印：水印算法和不可感知性测度[J]. 电子学报，2007，35(9)：1702-1705.

[6] 李跃强. DCT 域音频水印透明健壮算法[J]. 计算机工程与应用，2010，46(3)：84-86.

[7] 马翼平，韩纪庆. DCT 域音频水印：嵌入对策和算法[J]. 电子学报，2006，34(7)：1260-1264.

[8] 张金燕，赵占杰. 基于快速独立分量分析的音频盲数字水印算法[J]. 北京石油化工学院学报，2009，17(1)：28-31.

[9] 暴晋飞，柏森，朱桂斌，等. 基于能量比的小波域音频水印算法[J]. 计算机应用研究，2010，27(3)：1035-1038.

[10] 孙见青，汪荣贵，李守毅. 基于 DC 分量和 AC 分量相结合的数字水印技术[J]. 合肥工业大学学报：自然科学版，2007，30(7)：825-828.

[11] 黄雄华，王宏霞. 一种 DCT 域自适应音频水印算法[J]. 计算机应用研究，2009，26(8)：2989-2991.

（上接第 3757 页）

[11] GOODELL G, AIELLO W, GRIFFIN T, *et al*. Working around BGP: an incremental approach to improving security and accuracy of interdomain routing[C]//Proc of the Network and Distributed System Security Symposium. 2003: 75-85.

[12] KARLIN J, FORREST S, REXFORD J. Pretty good BGP: improving BGP by cautiously adopting routes[C]//Proc of IEEE International Conference on Network Protocols. 2006: 290-299.

[13] SUBRAMANIAN L, ROTH V, STOICA I, *et al*. Listen and whisper: security mechanisms for BGP[C]//Proc of Symposium on Networked Systems Design and Implementation. 2004: 29-31.

[14] SIGANOS G, FALOUTSOS M, FALOUTSOS P, *et al*. Power laws and the AS-Level Internet topology[J]. IEEE/ACM Trans on Networking, 2003, 11(4): 514-524.

[15] GAO Li-xin. On inferring autonomous system relationships in the Internet[J]. IEEE/ACM Trans on Networking, 2001, 9(6): 733-745.

[16] GE Zi-hui, FIGUEIREDO D, JAISWAL S, *et al*. On the hierarchical structure of the logical Internet graph[C]//Proc of SPIE ITCom. 2001: 208-222.

[17] University of Oregon route views project[EB/OL]. http://www.routeviews.org/.

[18] AGER B, CHATZIS N, FELDMANN A, *et al*. Anatomy of a large European IXP[C]//Proc of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. 2012: 163-174.

[19] PeeringDB [EB/OL]. [2013-04-20]. https://www.peeringdb.com/private/index.php.

[20] MOHAPATRA P, SCUDDER J, WARD D, *et al*. RFC 6811, BGP prefix origin validation [S]. 2013.

[21] KONG Ling-jing, ZENG Hua-xin. Use of distributed trustworthy node to secure AS_PATH[C]//Proc of International Conference on Multimedia Information Networking and Security. 2012: 35-38.