学士学位论文

面向 AAC 音频的安全隐写算法研究

学　　号：　20151002497

姓　　名：　刘婷

学 科 专 业：　信息安全

指 导 教 师：　许瑞　讲师

培 养 单 位：　计算机学院

二〇一九年六月

中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《面向 AAC 音频的安全隐写算法研究》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：

日　期：　　年　　月　　日

摘 要

数字媒体中的隐写术作为一种隐蔽通信的方式，一直以来为信息安全研究人员所关注。随着数字音频编码技术的发展，高级音频编码（AAC，Advanced Audio Coding）已经向各类应用领域渗透，如 iTunes、YouTube、腾讯音乐等。AAC 音频被广泛使用的同时，也为隐写术带来了丰富的载体空间，因此面向 AAC 音频的安全隐写算法意义重大。不可感知性和不可检测性（抗隐写分析性）是隐写术的安全目标，基于此，本文的主要研究内容如下：

1）深入探讨 AAC 编解码原理中的量化模块和无噪编码模块，为研究 AAC-Qmdct 隐写算法（Qmdct：量化后的 MDCT 频谱系数。AAC-Qmdct：以 Qmdct 为嵌入域的 AAC 音频）奠定知识基础。钻研心理声学模型模块原理和最小化失真理论（STC 编码），为设计隐写失真函数提供理论依据。学习 PEAQ（感知音频质量评价模型）工具的使用、掌握机器学习知识以及相关工具的使用，为实验评估工作做好技术储备。

2）根据最小化失真隐写理论和人耳听觉掩蔽效应，设计了基于心理声学模型的失真代价函数，实现了基于 STC 编码（STC，Syndrome-Trellis Codes）的 AAC-Qmdct 隐写算法。实验结果表明，本文提出的方案在隐写容量、不可感知性方面表现良好，不可检测性有待提高。

3）基于 python tkinter 界面设计技术，实现了一款 AAC 音频隐写系统。集成本文所提出的基于 STC 编码的 AAC-Qmdct 隐写算法，实现了音频载体导入、秘密信息文件导入、嵌入率设置、失真函数设置、消息嵌入、消息提取等功能。

本文根据 AAC 编码当中的心理声学模型知识设计了失真函数，提出了基于 STC 编码的 AAC-Qmdct 隐写算法。实验表明，本算法可以保证不可感知性。但是不可检测性能比较弱，所以未来工作方向是设法提高抗隐写分析性。此外，本文实现了一款 AAC 音频隐写系统，可以为互联网上的隐蔽通信（如传送军事情报等）提供有效的技术支持。

关键词：AAC；量化 MDCT 系数；隐写；STC；心理声学；失真函数

Abstract

Steganography in digital media is an important means of covert communication. With the development of digital audio coding, AAC (Advanced Audio Coding) has been widely used (such as YouTube, QQ music) thus bringing rich carrier space for steganography. Imperceptibility and undetectability are the security targets of steganography, so the security steganography algorithm for AAC audio is of great significance. Given all that above, the main contents of this paper are as follows:

1) Quantization and noiseless coding in the AAC codec were deeply discussed to study the AAC-Qmdct steganography algorithm (Qmdct: quantized MDCT spectral coefficient. AAC-Qmdct: use Qmdct as the embedded domain in AAC audio). Psychoacoustics and the theory of minimizing embedding impact using STC (Syndrome-Trellis Codes) are studied to provide theoretical basis for designing distortion function. To offer technical support for the experiment evaluation, it's a must to learn PEAQ and machine learning tools.

2) According to STC and hearing threshold theory, this paper designed a distortion function based on psychoacoustics and the AAC-Qmdct algorithm based on STC. Experiments show that the method proposed performs well in steganographic capacity and imperceptibility, but not well in undetectability.

3)This paper implemented an AAC audio steganography system based on the python tkinter interface technology. Integrating the proposed algorithm, the system realized many functions, including file import, embedding rate setting, distortion function setting, message embedding and message extraction.

In this paper, the distortion function based on psychoacoustics and the AAC-Qmdct steganography algorithm based on STC is proposed. Experiments show that it cannot guarantee undetectability but imperceptibility, so the future work is to improve the latter. In addition, the AAC audio steganography system this paper implements can provide effective technical support for covert communication on the Internet.

Key words: AAC; Quantized MDCT spectral coefficient; Steganography; STC; Psychoacoustics; Distortion function

目 录

第一章 绪论

1.1 问题背景

隐写术是一种隐蔽通信的手段，目的在于隐藏消息的存在性，即敌手不知道传输的文件中是否含有秘密信息。实现途径是：在载体文件（cover）上以某种隐写算法嵌入秘密信息得到载密文件（stego），通过发送 stego 来完成秘密信息的传输，这整个过程是不可感知的。图像、音频、视频等多媒体文件具备大量的冗余，而这些不会被人类视觉、听觉感知的冗余恰好可以用来嵌入秘密信息[1]。因此，很多情况下，多媒体文件会被作为隐写载体的不二之选。

随着数字音频编码技术的发展，压缩音频的使用已经渗透进人们的日常生活。在早期，以 MP3 为代表的压缩格式的音乐在互联网上风靡一时。到了 20 世纪末 AAC 标准被提出，人们逐渐意识到 AAC 音频的价值优势：与传统的 MP3 相比，AAC 在相同码率下音质表现更好。AAC 音频的应用面得以推广，如苹果公司的数字媒体播放应用程序 iTunes、视频分享平台 YouTube、腾讯旗下的 QQ 音乐应用等。

AAC 音频的广泛使用，同时为隐写术提供了丰富的载体空间。隐写术最早应用于图像领域，近年来音频隐写的研究热度逐渐上升。目前已经出现多种基于不同压缩标准的隐写算法，例如面向 MP3（MPEG-1 Layer III）、AAC 音频的隐写算法。但是，现有的方法在方案设计上缺乏理论支持，安全问题（如不可感知性、不可检测性）上没有做到全方位的保证，更加安全的 AAC 隐写方法亟待提出。因此，面向 AAC 音频的隐写研究具有深远的研究意义。

STC 编码（STC，Syndrome-Trellis Codes）作为一个优秀的基于最小化失真理论的隐写框架，已经在图像、音频隐写工作中取得了一定的成果。失真代价函数的设计是使用 STC 框架的关键。因此，本文使用 STC 编码隐写框架，根据心理声学模型知识来设计失真代价函数，旨在实现面向 AAC-Qmdct 的安全隐写方法。

1.2 研究现状

MP3 与 AAC 编码原理一脉相承，因此现有的针对 MP3 的隐写算法对 AAC 音频的隐写研究很有借鉴意义；失真函数的设计是应用 STC 隐写框架的关键，怎样设计失真函数来保证安全性是个重要问题。基于以上两点考虑，本文有针对性地进行了分析调研，主要分为两个方面：MP3、AAC 音频频域隐写的研究现状和 STC 编码中失真函数设计的研究现状。

1.2.1 MP3、AAC 频域隐写研究现状

目前针对 MP3 频域的隐写算法研究成果较多，例如公开的 MP3 隐写工具 Under MP3 Cover[2]和 MP3Stego[3]等。Under MP3 Cover 是通过修改全局增益的 LSB 最低位比特来实现隐写的，很容易被检测；MP3Stego 则是在量化模块的内循环中，改变音频帧长度的奇偶性来实现隐写，虽然不易被检测，但是嵌入率太低，每个颗粒最多只能嵌入一个比特（MP3 中每一帧是 2 个颗粒，每个颗粒有 576 个样本点）。

在 AAC 音频隐写工作中，依据其编码特性可以分为四个嵌入域：比例因子域、MDCT 系数域、Huffman 编码域，如图 1.1 所示。

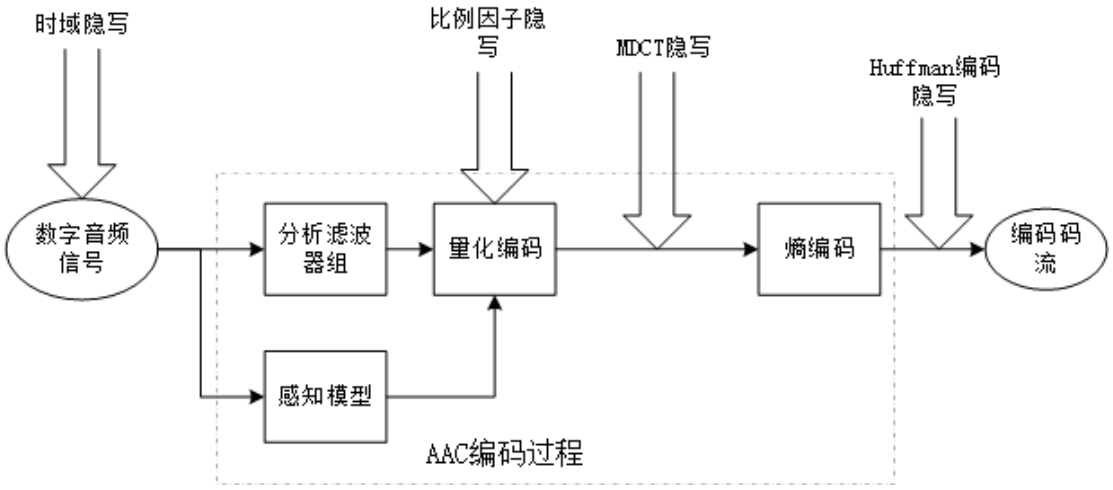


图 1.1 AAC 音频隐写域

对于比例因子域，中国科学技术大学的魏一方根据 MP3Stego 的设计思路，提出了基于码率控制[4]的 AAC 隐写方法。由于方法也是基于帧长度的奇偶性

来实现隐写的，一帧（1024 个样本点）同样是嵌入一个比特。嵌入负载（秘密消息比特数/载体长度）= 1/1024。

对于 MDCT 系数域，考虑到符号位数值为 0 或 1，通过修改符号位的数值来设计隐写规则是可行的。直接对符号位进行修改对 MDCT 系数的改动较大，因此可以选取符号位数值的异或值[5]作为秘密信息进行嵌入。低频部分的 MDCT 系数幅度较大（能量主要集中在低频），中高频部分的 MDCT 系数幅度较小（多数为-1、0、1）。接近高频的 MDCT 系数部分本身就具有较大量化误差，轻微改动这一部分对音频质量的影响是不明显的，因此可以设计一种基于 Qmdct 小值区系数的隐写方法[6]，通过修改 quant[3]来实现隐写（在 Huffman 编码之前的 Qmdct 被分组，一组 m 个。m=4 时，quant[3]是一组中的最后一个 Qmdct）。以上都是直接修改 MDCT 系数的方式，此外还有间接修改的方式，如 QIM[7]方法：量化时，通过修改比例因子（即选用不同的量化器）来控制当前比例因子带中 MDCT 系数的量化区间，同时保证量化后 MDCT 系数值所在的区间与秘密信息形成的有映射关系，从而来实现秘密信息嵌入。在提取秘密信息时，通过判断指定位置的 MDCT 系数值所在的范围即可判断当前位置嵌入的秘密信息。

1.2.2 STC 编码中失真函数设计研究现状

校验格码（STC，Syndrome-Trellis Codes）是一种最小化嵌入失真的通用隐写框架[8]，最早由 Tomáš Filler 于 2010 年提出。它是矩阵编码的一种，广泛用于自适应嵌入隐写。这种方法最大的优点是，对于需要嵌入信息的发送端，只需定义失真函数即可。对于需要提取信息的接收端，只需要得到消息长度，无需知晓失真函数。该方案落地的应用最早出现在图像隐写研究上，后来慢慢渗透进音频隐写领域。

目前出现了一些利用 STC 框架来实现音频隐写的方法。中山大学的骆伟祺提出的方案[9]是：以单个 WAV 音频信号（即 AAC 时域信号）作为嵌入单元，根据原始 WAV 音频，和经过一次 AAC 编码再解码的 WAV 音频的信号差值设计每个嵌入单元修改的失真代价，并利用 STC 框架自适应嵌入信息并实现正确提取。设置秘密信息的嵌入率 ratio（原始 WAV 音频平均每个样本修改的比特数，单位为 bps），则嵌入的秘密信息长度 m= [n*ratio]，n 是原始 WAV 音频样本总

数。依据设定好的失真代价及需嵌入的 m 比特秘密信息对选定的载体做 STC 操作，最小化所有样本的修改代价，实现自适应嵌入。这种方法比较启发式，是利用残差序列经验值来构造失真函数的。不足之处是方案构造中没有利用到人耳的听觉感知特性，且工程实现上编解码计算量过大。

骆伟祺的另外一个方法是选择 MP3 频域信号作为嵌入域[10]，通过比较单个的 Qmdct 的量化失真 DQ 和该系数所在的比例因子带的失真 error_energy 的大小关系来设计失真函数。如果 DQ 大于 error_energy，则赋予该 Qmdct 系数以一个较大的失真代价，比如可以是帧的长度；否则赋予该 Qmdct 系数以一个较小的失真代价，比如 1。这种设计判断条件的思路来源于 MP3（MPEG-1 layerⅢ）编码中的量化失真，而量化模块的外循环（详见 2.1.3）正是根据人耳对音频的听觉感知特性构造而成的。此方法在不可感知性和不可检测性方面表现良好。但是隐写容量太低，一个颗粒（576 个样本点）最多嵌入 10 比特，即负载（秘密消息比特数/载体长度）≈1.7%。

中国科学院信息工程研究所的易小伟等人提出的方法[11]是：选择 MP3 音频的 Huffman 码字为嵌入域，通过修改 Huffman 编码域，从而间接地修改了 Qmdct 系数（详见 2.1.4）。提出的失真函数设计方法用到了心理声学模型中的安静阈值函数曲线，该曲线和人耳听觉感知特性密切相关，所以在不可感知性方面表现很好。

综上所述，要想利用 STC 框架在 AAC 频域上做隐写，如何定义失真函数，是最关键的问题。

1.3 论文主要内容及组织结构

本文首先对 AAC 编解码原理、最小化失真嵌入理论（STC 编码）进行了深入了解，从而提出了基于 STC 编码的 AAC-Qmdct 的隐写算法，并实现了一款 AAC 音频隐写系统。本文的整体结构如下：

第一章，绪论部分。提出问题，说明隐写术的初衷，解释选择 AAC 音频来做隐写研究的意义。鉴于 MP3 隐写对 AAC 隐写具有借鉴意义，以及 STC 编码的优良特性，本文有针对性地介绍了研究现状。

第二章，介绍与本次 AAC 音频隐写研究相关的技术背景，包括感知音频编码，AAC 编码，隐写与隐写分析。重点描述 AAC 编码中的心理声学模块、量化模块、STC 隐写框架，为本次隐写方案中的失真函数设计提供理论支持。

第三章，本文的核心部分。为了在 AAC 音频上完成隐写工作，本章描述了如何设计失真函数，应用到 STC 隐写框架中。首先对现有的失真函数设计方法进行调研，分析利弊。总结函数设计的思路，提出一种新的根据量化失真从而构造隐写失真的方案，并与现有方案进行对比。实验证明，本方案不可感知性良好，不可检测性欠佳。

第四章，展示工程实现的效果。完成了一个 AAC 音频隐写的软件系统，实现了文件导入、消息嵌入、消息提取等功能。

第五章，总结全文研究内容，针对现有工作存在的问题，描述未来工作开展的方向。

第二章 相关技术背景

本章介绍了与 AAC 音频隐写研究有关的技术背景，包括 AAC 编解码原理，STC 最小化失真嵌入理论，隐写分析相关知识。在介绍 AAC 编码时，详细描述了量化模块和无噪编码模块，为研究 AAC-Qmdct 隐写算法奠定知识基础；详细阐释了心理声学模型模块原理，为 STC 编码预备工作（设计隐写失真函数）提供理论依据。

2.1 AAC 编码原理

2.1.1 AAC 编码概述

音频信号编码指信源编码，即数据压缩。如果未经过数据压缩，直接量化进行传输则被称为 PCM（脉冲编码调制）。最为广泛使用的感知音频编码[12]利用心理声学模型，采用尽可能低的数据率获得信源输出的感知无失真表示。在相同比特率的情况下，感知编码能得到更高的感知音频质量。利用人耳的感知模型，去除声音信号中人耳不能感知的部分，以达到降低数据率的目的。基本框架如下图。

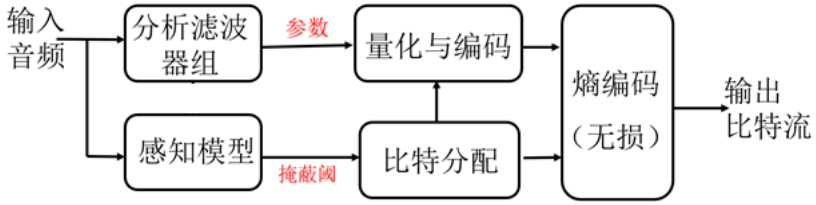


图 2.1 感知音频编码框架

MPEG-1 是最早的针对高质量音频的感知编码的第一个国际标准，分为 I、II、III（即我们熟知的 MP3）；MPEG-2 分为 MPEG-2 BC（backward compatibility，后向兼容）、MPEG-2 NBC（非后向兼容，如 AAC 标准）。后向兼容是，按照旧的编码标准设计成的解码器，也能对按照新的编码标准产生的比特流进行解码。

MPEG-2 AAC（Advanced Audio Coding）是感知音频编码的一种[13]，其采样率是 8 - 96 kHz，码率是 8-128 kbps（单声道）或 16-256 kbps（立体声）。码率是经过压缩编码后的音频数据每秒钟传送的比特数，单位一般采用的是 kbps。音

频采样当中有一个著名的奈奎斯特采样定理：采样率 fs 必须大于被测信号感兴趣最高频率分量 fn（又称奈奎斯特频率）的 2 倍，原波形信号才能完整重建；否则会出现混叠现象，即采样数据中出现虚假的低频成分。AAC 规定了 2 种封装格式 ADIF 和 ADTS，前者用于属于文件格式用于存储，后者属于流格式，用于传输。

AAC 编码原理如图 2.2（引自 AAC 编码[13]技术标准）。AAC 解码可以理解为将图 2.2 中的输入输出位置、箭头方向全部调换之后得到的流程图，即与 AAC 编码相对应的逆过程。

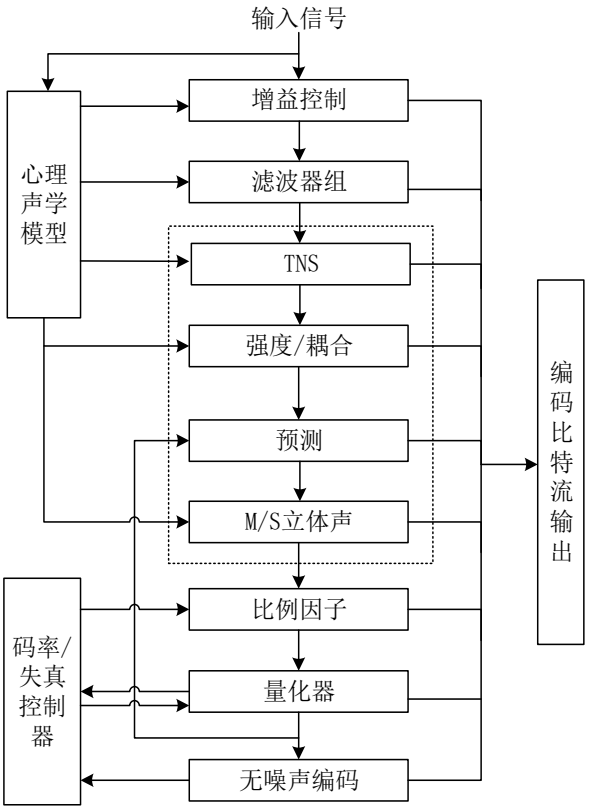


图 2.2　AAC 编码流程图[13]

音频信号经过滤波器组模块以后从时域变换到频域，由 MDCT 变换完成。音频信号在经 MDCT 转换前，需经过一个窗函数（window function）相乘。作

用是抑制频谱泄漏（频谱泄漏指的是，长序列信号截短后，造成谱峰下降、频谱扩展的现象）。变换长度有长变换（1024 点）和短变换（128 点）。对于长变换，比例因子频带（scale factor band，sfb）和比例因子窗口频带（scale factor window band，swb）是等同的。对于短变换，共享一套比例因子的几个（8 个以内）比例因子窗口频带可以看作一个比例因子频带。一个比例因子频带指的是 4 个或者 4 个以上的频谱系数的集合，它们共享相同的比例因子。比例因子频带宽度的确定模仿人类听觉系统的临界频带（critical bandwidth）。其个数和宽度取决于变换长度和采样率。后续章节重点描述心理声学模型、量化模块、无噪编码模块。

2.1.2 心理声学模型模块

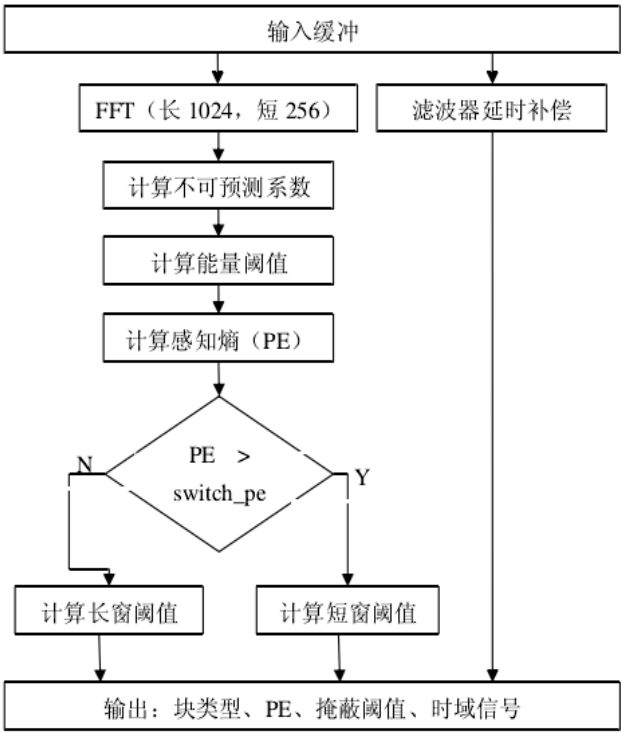


图 2.3 心理声学模型流程图[13]

AAC 中，心理声学模型如图 2.3 所示。图中，输入是时域信号（与滤波器组并行工作），输出的重点是掩蔽阈值 Xmin（各个比例因子频带的最大可允许失真）、由心理听觉熵 PE 计算的比特分配 more_bits。这两者会作为量化模块（quantization）的输入。

2.1.3 量化模块

量化包含三重循环：帧循环（调用外循环）、外循环（调用内循环）和内循环。

帧循环功能是计算可用比特数和初始化迭代变量。每帧编码需要的比特数，部分取决于 more_bits（心理声学模型模块计算得出）；

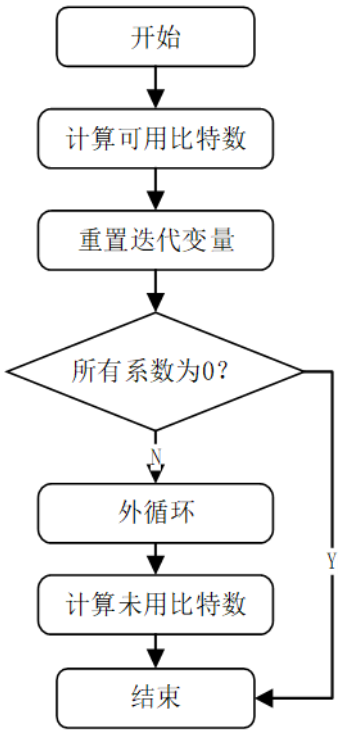


图 2.4 帧循环流程图

外循环控制量化失真，若比例因子频带 sb 的失真 error_energy(sb)超过最大可允许失真阈值 Xmin(sb)（心理声学模型模块计算得出），则比例因子进行放大，重新进入内循环进行量化编码（mdct_line(i)是实数，x_quant(i)是整数）。关键部分伪代码如下。

1. **do for** each scalefactor band sb:
2. error_energy(sb)=0
3. **do** from lower index to upper index i of scalefactor band

4. error_energy(sb) = error_energy(sb) + (abs(mdct_line(i))- (x_quant(i) ^(4/3) * 2^(¼ * (scalefactor(sb) -common_scalefac))))^2
5. end **do**
6. end **do**
7. **do for** each scalefactor band sb
8. **if** (error_energy(sb) > xmin(sb)) then
9. scalefactor(sb) = scalefactor(sb) - 1
10. end **if**
11. end **do**

根据 AAC 量化模块中的外循环（失真控制）部分伪代码可知，步骤 4 中，$(|mdct_line(i)| - x_quant(i)^{\frac{4}{3}} \times 2^{\frac{1}{4}\times(scalefactor(sb)\ -common_scalefac)})^2$便是单个量化系数的量化失真。

其中平方括号里的第二项就是反量化，可以理解为$inv(x_quant(i)) = x_quant(i)^{\frac{4}{3}} \times 2^{\frac{1}{4}\times(scalefactor(sb)\ -common_scalefac)}$。

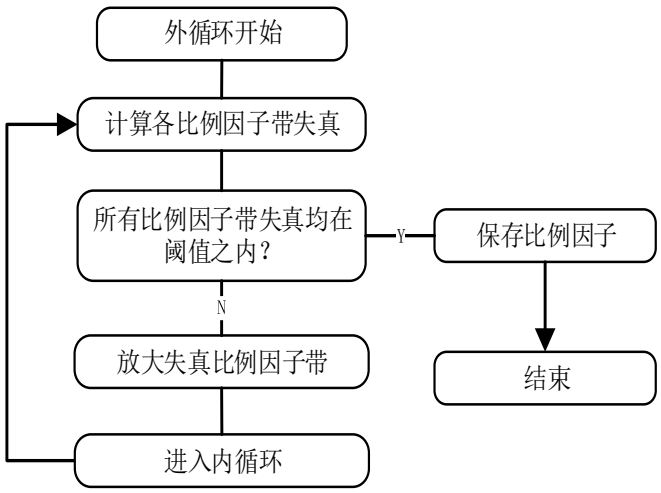


图 2.5 外循环流程图

内循环用于码率控制（量化编码比特数），如果编码所需比特数超出了可用比特数，则通过修改量化步长的方法使得编码比特数在可用比特数的范围内。其中，无噪编码（Huffman 编码）在内循环中进行。

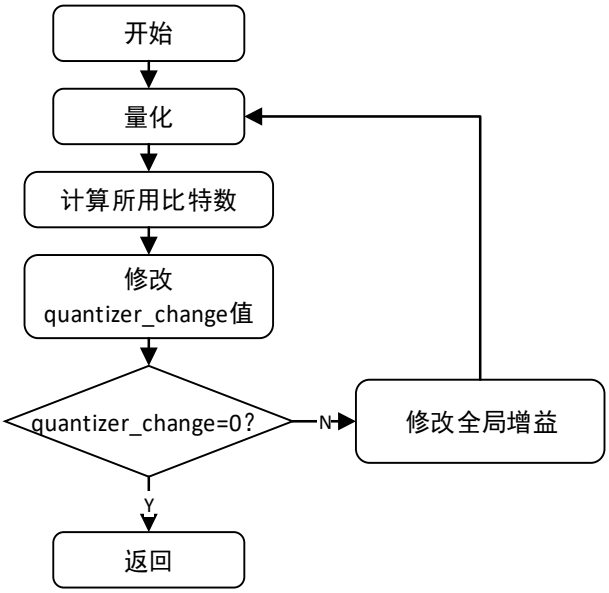


图 2.6 内循环流程图

2.1.4 无噪编码

在量化模块的内循环中，采用 12 个固定码书对 Qmdct（量化以后的 MDCT 频谱系数，简称 Qmdct）以及比例因子等其它参数进行无噪编码，进一步除去数据冗余。过程包括分区、分组和交叉、比例因子编码、Huffman 编码。

（1）分区

为了使多个比例因子带共用一个码书，增加压缩效率，分区过程（贪心算法）将每一个比例因子带划分为一个区，且每个分区采用的码书序号尽可能小，使得比例因子带编码后比特数最少。对分区进行两两合并，若合并后的码书编码所需要的比特数更少，则合并；否则分区保持不变。 重复以上过程，直到分区不能再被合并为止，如图 2.7。

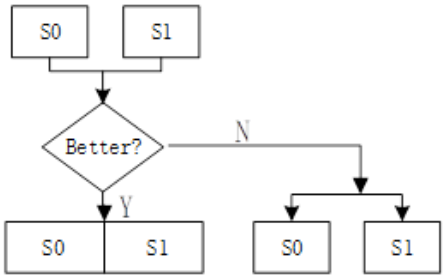


图 2.7 比例因子带分区流程图

（2）分组与交叉

为了在编码短块时，尽可能的减少比例因子的编码数量，这里忽略频谱系数所在的块，将连续的、具有相同比例因子的频谱系数分为一组，共享一个比例因子。

分组前，Qmdct 可表示为 q[g][win][sfb][bin]，其中 g 为分组索引，win 为块索引，sfb 为比例因子带索引，bin 为频谱系数索引。分组后，Qmdct 可表示为 q[g][sfb][win][bin]。分组交叉前后，比例因子带的频谱顺序如图 2.8-2.9 所示。

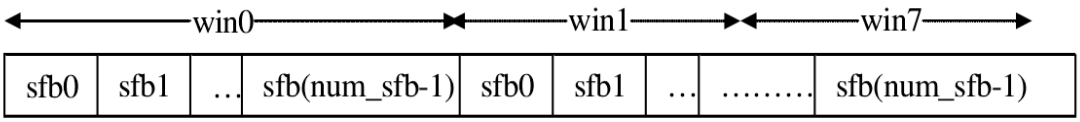


图 2.8 比例因子带的频谱顺序[13]（分组交叉前）

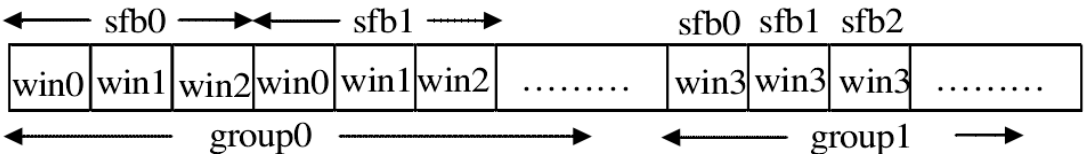


图 2.9 比例因子带的频谱顺序[13]（分组交叉后）

（3）比例因子编码

全局增益被编码 8 位的无符号整数。将全局增益作为第一个比例因子，计算比例因子之间的差值，对差值进行 Huffman 编码。

（4）Huffman 编码

Huffman 编码属于无损编码，编码对象是 Qmdct、比例因子等参数。编码原理是将 m 个 Qmdct 编成一个 Huffman 码字，其中 m 依据码书的不同分别为 2 或者 4。以 m=4 为例，index = 27*quant[i] + 9*quant[i+1] + 3*quant[i+2] + quant[i+3] + 40。算出 index 后，根据 Huffman 编码码书，查询即可得到码字。

AAC 编码标准中规定的 12 本固定码书如下表 2.1 所示，我们可以观察到每个码书对应的 Qmdct 范围以及是否有符号等情况。无符号的码书指的是参与编码的 Qmdct 为原始系数的绝对值，符号位需要加在码字后进行传输。0 号码书适用于 Qmdct 均为 0。11 号码书适用于 Qmdct 大于 16 的情形。

表 2.1 Huffman 编码码书

码书序号	数据个数 m	最大绝对值	有符号值
0	/*	0	/*
1	4	1	是
2	4	1	是
3	4	2	否
4	4	3	否
5	2	4	是
6	2	4	是
7	2	7	否
8	2	7	否
9	2	12	否
10	2	12	否
11	2	16(ESC)*	否

2.2 STC 编码

由于用于隐写的多媒体载体文件被修改后常常难以维持本身的统计分布特性，失真较大，为了解决这个问题，校验格码（STC，Syndrome-Trellis Codes）应运而生，目标在于在最小化嵌入失真的条件下，实现消息的正确嵌入和提取。

STC 是一种通用隐写框架，由 Tomáš Filler 于 2010 年提出[8]。解决问题的一般思路是在嵌入率一定的情况下使得嵌入失真最小化，或者是在失真一定的情况下使得嵌入率最大化，前者使用的居多。校验格码是矩阵编码的一种，用到卷积、最小化嵌入失真的知识，广泛用于自适应嵌入隐写。嵌入效率（嵌入载体的秘密消息数/载体失真量）表现尤佳。这种方法最大的优点是，对于需要嵌入信息的发送端，只需定义失真函数即可。对于需要提取信息的接收端，只需要得到消息长度，无需知晓失真函数。

该编码方案目标是构造一种方法，使得在 D(X, Y)最小化的前提下，实现 m 比特消息正确的嵌入和提取。主要思路如下：

设载体样本$X = \{x1, x2, \ldots xi \ldots xn\}$，载密样本$Y = \{y1, y2, \ldots yi \ldots yn\}$，嵌入消息数是 m 比特，n 是图像像素个数、音频帧中的采样点个数等等，负载$\alpha = m/n$

（要保证$\alpha \leq 0.5$）。x_i 被修改为 y_i 的失真代价为$\rho_i(y,x)$（加性失真），失真代价累加和为 D(X,Y)，如（式 2.1）。

$$D(X,Y) = \sum_{i=0}^{n} \rho_i (y_i, x_i) \quad （式 2.1）$$

STC 的理论基础是矩阵编码。在矩阵编码中，设发送端和接收端共享一个 m×n 维的奇偶校验矩阵$H \in \{0,1\}$。发送端要做的是寻找能够代替 x 的 y，使得 D(X,Y)最小，同时使得消息能够被提取，即 Hy=m。接收端的工作很简单，计算 Hy 即可得到秘密消息 m。所以，发送端只需要定义好失真函数，接收端只需要知道消息长度，无需知晓失真函数。如式（2.2-2.3）。

$$y = Emb(x,m) = \arg\min||_{y=m} D(x,y) \quad （式 2.2）$$

$$m = Ext(y) = Hy \quad （式 2.3）$$

不幸的是，如果使用一般构造的矩阵 H，这个嵌入问题难度是 NP 的。为了解决这个问题，STC 方案[8]提出用伪随机子矩阵 H'的方式来构造 H，使用 Viterbi 算法作为优化工具寻找最短路径，从而找到最合适的载密对象 y。

校验格码最早应用于图像隐写研究，后来慢慢渗透进音频隐写领域。比如，在图像中，可用纹理复杂度定义失真函数，对于同样的修改幅度，纹理越复杂的区域造成的失真值越小。因此，要想有效地将 STC 框架应用到其他领域中，如何定义失真函数，是最关键的问题[14]。

2.3 隐写分析

2.3.1 隐写分析目标

作为与隐写术相对立的技术，隐写分析的目的在于检测秘密消息的存在性。有别于密码分析，破解隐写系统一般不需要读取秘密消息，当然在某些应用场景中是需要读取秘密消息的，这项任务属于取证隐写分析。在囚犯问题中，Alice 和 Bob 允许进行通信，但是看守者 Eve 会监视隐写信道，Eve 的行为被称为隐写分析。如果隐写分析者能够以高于随即猜测的概率（0.5）来区分载体和载密对象，那么她对隐写信道的攻击就是成功的，即隐写系统被破解了。

隐写分析包括针对性隐写分析和盲隐写分析[15]。针对性隐写分析是假设隐写分析者已知隐写方法或者已知嵌入域，有针对性地提取特征，采用隐写分析器（比如传统机器学习中的 SVM）进行分类；盲隐写分析在技术实现上与之类似，不同之处是隐写分析者不知道消息是用什么方法嵌入的或者是什么地方嵌入的，这时候就需要提取大量的特征集。本文实验评估不可检测性时，使用的是针对性的隐写分析，详见 3.3.3。

隐写分析分为两步。一是特征提取，即从 cover 和 stego 集中提取隐写分析特征。二是将提取出的特征放入学习器（隐写分析器）里以分类 cover 和 stego，这里包括有训练、测试过程。在第二步中，常常用到机器学习的概念工具，下面对此进行一些介绍。

2.3.2 学习器的性能度量

设定载密对象为正例，载体对象为反例，数据集 D 是已经提取到的载体和载密对象的特征。现将 D 分为训练集 S 和测试集 T，它们一般是互斥的。学习器学到的模型适用于新样本的能力，称为泛化性能。评估泛化性能（这里理解为隐写分析的效果）的衡量标准有，检测率（又称 accuracy）, ROC，AUC 等。

对于二分类问题，分类结果混淆矩阵如下表 2.3。

表 2.3 分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

对于检测率 Pd，采用真正例率（True Positive Rate，TPR）和真反例率（True Negative Rate，TNR）来表示，具体计算方法如式 2.4-2.6 所示：

$$\mathrm{TPR} = \frac{TP}{TP+FN} \quad \text{（式 2.4）}$$

$$\mathrm{TNR} = \frac{TN}{FP+TN} \quad \text{（式 2.5）}$$

$$P_D = 1 - \frac{P_{MD}+P_{FA}}{2} = \frac{TPR+TNR}{2} \quad \text{（式 2.6）}$$

ROC 全称是受试者工作特征（receiver operating characteristic，简称 ROC），用于衡量学习器的效果。学习器调整分类阈值（每个样例的预测值），依次把每个样例划分为正例。得到真正例率（True Positive Rate，$TPR = TP/(TP + FN)$ ）和假正例率（False Positive Rate，$FPR = FP/(TN + FP)$）坐标对，从而绘制成 ROC 曲线。该曲线是凸函数，如果一个学习器的曲线“包住”另一个曲线，说明前者的性能比后者好[16]。如果存在交叉，则可以比较曲线下的面积（AUC）。下图是有限个样本下的 ROC 曲线。

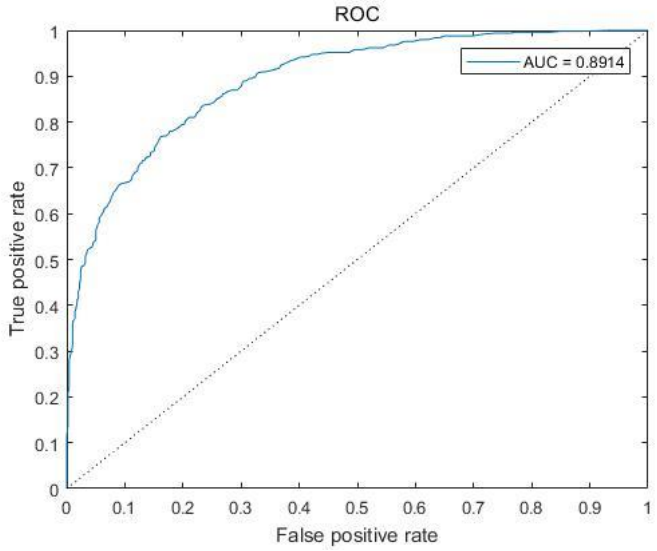


图 2.10 ROC 曲线

2.4 本章小结

本章介绍了与 AAC 音频隐写研究有关的技术背景。在阐释 AAC 编码时，详细描述了量化模块、无噪编码模块和心理声学模型模块，为下一章中的失真函数设计提供理论依据。接着描述了最小化失真隐写理论（STC 编码），为研究基于 STC 编码的 AAC-qmdct 隐写算法做好知识铺垫。最后介绍了与隐写分析工作相关的机器学习技法，为下一章评估不可检测性做好技术储备。

第三章 基于 STC 编码的 AAC 频域隐写方案

本隐写方案选择 AAC 频域（即 AAC-Qmdct）为嵌入域，利用音频量化、心理声学模型知识构造失真函数（根据上一章节的技术背景知识介绍我们了解到，要想有效地利用 STC 框架，如何定义失真函数，是最关键的问题），使用 STC 隐写编码框架实现安全的 AAC-Qmdct 隐写算法。在实验评估环节，本文通过检测 AAC 音频在隐写前后的音质变化来度量不可感知性，通过特征提取、集成分类器训练测试（对载体和载密进行分类）来度量不可检测性，与现有方法进行对比分析，衡量综合表现效果。

3.1 失真函数设计

3.1.1 现有方法思路

合理构造失真函数是应用 STC 隐写编码的重难点。鉴于 MP3（MPEG-1 layer Ⅲ）和 MPEG2-AAC 的量化方式一脉相承，我们借鉴 MP3 频域隐写的思路，来设计本次 STC 隐写方案中的失真函数。

元素 x 被更改为 y 的代价是 ρ，累加和为失真 D。这是 Filler 提出 STC 框架时候对隐写失真函数进行的较为笼统的定义。

第一种常见的思路是 $\rho=(x-y)^2$。即 ρ=1（元素±1），ρ=4（元素±2），ρ=∞（元素 255 加 1、元素 0 减 1）。

第二种是由量化失真从而推出隐写失真。考虑到单个量化系数修改引起的失真会影响该系数所在的整个比例因子频带的失真[10]，对于修改之后的 Qmdct，如果单个 Qmdct 的量化失真大于其所在的比例因子带的平均量化失真，则赋予该系数较大的隐写失真代价（比如帧长度），反之赋予较小的代价（比如 1，等等）。

第三种是利用心理声学模型知识。在感知音频编码中，频域信号量化的过程就是利用了心理声学模型的原理（如听觉掩蔽效应），省去不必要的信号传输，实现音频压缩。

听觉阈值（hearing threshold），或安静阈值，表示在给定频率下可以听到的最低声级。它对于音频编码非常重要，因为低于这一水平的信号中的频率分量我们

人耳是感知不到的，因此它们不需要被传输。安静阈值对应 phon = 3 所描述的等响度轮廓，近似公式如下。f 为频率,单位 kHz，听觉范围是 20 Hz~20 kHz。T(f)是为 f 频率下听觉的绝对阈值，对应的是声压级 SPL，单位 dB。

$$T(f) = 3.64f^{-0.8} - 6.5e^{-0.6(f-3.3)^2} + 0.001f^4 \quad （式 3.1）$$

对于特定频率 f 的频谱系数，T(f)越小，说明人耳对该频率信号的敏感度越高，该频谱系数修改后引起的听觉失真就越大。基于此，易小伟等人提出将隐写所带来的听觉失真作为隐写失真代价[11]，公式如下。这里 xi 是量化系数，xi'是修改后的量化系数。 ε 是常数，保证分母大于 0。

$$costs(x_i) = \frac{|x_i' - x_i|}{\log_2(T(f_i)+\varepsilon)} \quad （式 3.2）$$

在 3.3 章节中，本文将使用此失真函数（STC-Tf）（取 xi'-xi=1，ε=6）作为实验评估的对照。

3.1.2 本文方法思路

结合以上思路，充分利用音频量化和心理声学模型的知识，本文设计的失真函数如下：

$costs(Q_{mdct})$ （式 3.3）

$$= \begin{cases} \left(R_{mdct} - inv(Q_{mdct}')\right)^2, & err\_energy(sb') > X_{min}(sb) \ and \ Q_{mdct} \neq 0 \\ 0.01, & err\_energy(sb') \leq X_{min}(sb) \ and \ Q_{mdct} \neq 0 \\ INF, & Q_{mdct} = 0 \end{cases}$$

Qmdct 是量化后的 MDCT 系数，Qmdct'是被修改的系数（即进行了±1 操作），Rmdct 表示实数 mdct 系数（量化前）。sb 是 Qmdct 所在的比例因子频带，error_energy(sb)是 Qmdct 所在的整个比例因子频带的量化失真，Xmin(sb)是掩蔽阈值。inv()是反量化过程（详见第二章描述的心理声学模块和量化模块）。

当量化系数不为 0 时，如果被修改的量化系数所在的比例因子带的量化失真 error_energy(sb')大于掩蔽阈值 Xmin(sb)，隐写失真如（式 3.3）第一种情况所示；反之隐写失真被赋予一个小值，如第二种情况所示。当量化系数为 0 时，隐写代价是无穷大（INF），即我们不允许在 0 值系数上做修改。

3.2 流程描述

本次隐写方案的嵌入域选择的是 AAC-Qmdct。因此，嵌入位置在 AAC 编码中的“量化器”之后，“无噪声编码”之前（详见 2.1.1 章节的图 2.2）。相应地，提取位置在 AAC 解码中的“无噪声解码”之后，“反量化器”之前。

嵌入端的流程是，首先计算 cover 波形文件的 Qmdct（相当于 2.2 章节中的 x）的失真函数 loss（AAC 编码端中进行）；loss 被送入 STC 编码端，根据负载生成随机消息 m，计算输出 S_qmdct（相当于 2.2 章节中的 y）；S_qmdct 再送进 AAC 编码端得到载密 AAC 文件 stego。

提取端的流程是，载密 AAC 文件 stego 送入 AAC 解码端，提取得到 de_Sqmdct（正常地，de_Sqmdct 理应和嵌入的 S_qmdct 相同）；de_Sqmdct 送入 STC 解码端，提取消息，验证消息提取的正确性。

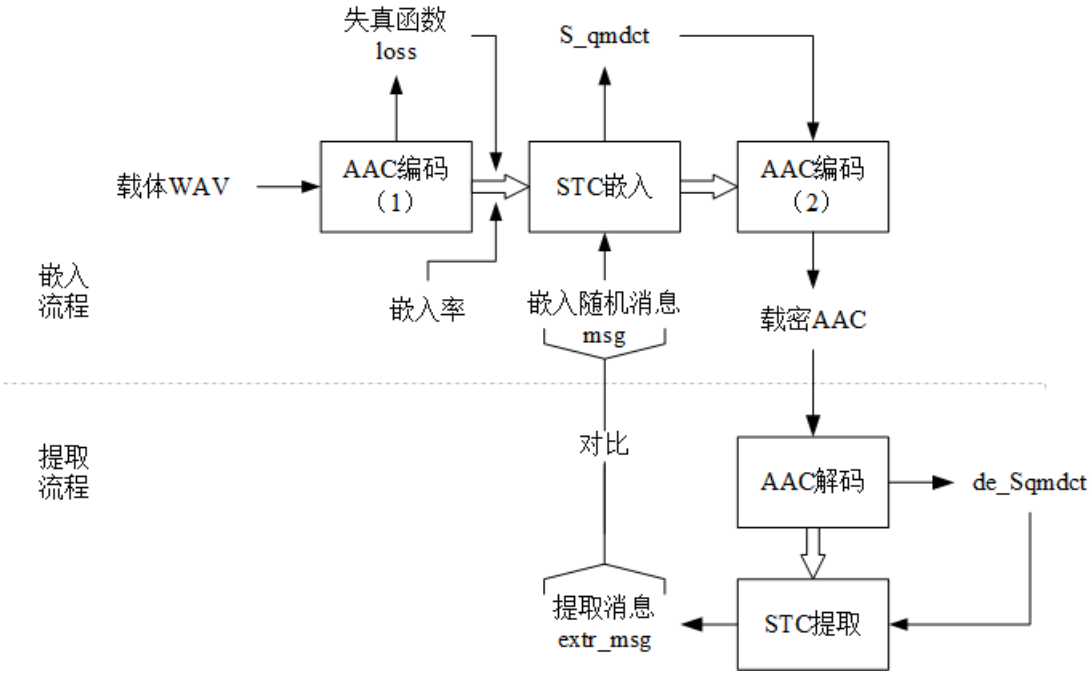


图 3.1 基于 STC 的 AAC-Qmdct 隐写流程（包括嵌入端、提取端）

3.3 实验评估

隐写算法的评估指标分别是隐写容量、不可感知性与不可检测性，三者存在相互制约的关系。隐写容量衡量的是载体样本中最多嵌入信息量的多少，不可感知性衡量的是隐写前后音频音质的差距，不可检测性衡量的是算法的抗隐写分析性能。在音质测量工作中，将原始 WAV 作为参照样本，将载密 AAC 文件解码后得到的 WAV 文件作为测试样本，得到客观失真等级 ODG。隐写分析工作中，在载体 AAC 文件、载密 AAC 文件上的 Qmdct 上提取特征（AAC 解码过程中进行），将这些特征送入集成分类器进行训练和测试，得出检测率 P_D。对于这三个指标，本文实验效果和现有的一些方法进行了对比分析。

3.3.1 隐写容量

本实验中，隐写容量衡量指标为负载α：$\alpha = m/n$，m 是秘密信息比特数，n 是 cover 的长度（这里是所有帧的所有 Qmdct 系数个数），这里单位定义为 bpc（bit per Qmdct_coefficient）。

对于隐写容量，本文提出的方法与现有的三个方法（详见章节 1.2）对比如下表 3.1。表格中的隐写方法，从上到下依次是：MP3Stego[3]、基于码率控制的 AAC 隐写[4]（简称 AACStego）、基于 STC 编码的 MP3 频域隐写方法[10]（简称 MP3-STC-Luo）、本文提出的方法（Proposed method，详见章节 3.1.2）。由表格数据可知，本文方法具有较大的隐写容量。

表 3.1 不同隐写方法的隐写容量

隐写算法	隐写容量（bpc）
MP3Stego[3]	0.00174
AACStego[4]	0.00097
MP3-STC-Luo[10]	0.01736
Proposed method	0.12500

3.3.2 不可感知性

音频文件大概分为语音文件和音乐文件。对于语音，使用 PESQ 主观语音质量评估来衡量。对于音频失真衡量，常用 PEAQ 感知音频质量评价模型[17]，该

模型由国际电信联盟无线电通信组（ITU-R）提出。PEAQ 标准适用于采样频率为 44.1-48kHz 的高质量音频信号，通过心理声学模型模拟人耳对音频信号的感知。根据参考信号（原始音频）和测试信号 （隐写后的音频）计算得到客观失真等级(Objective Difference Grade, ODG)：ODG 最高为 0，表示没有失真；最低为-4，表示失真很大。

本实验中，音频样本均采用的是 WAV 文件：参照样本、测试样本各 100 个，双声道，采样率 48 kHz，时长 5s。类别包括歌曲、演奏曲，风格包括舒缓型、嘈杂型等。参照样本是原始 WAV 文件，测试样本选择的是将载密 AAC 文件解压后得到的 WAV 文件。

不同失真函数，不同负载下，实验结果得到的 ODG 值如下表 3.2。横向是两种失真函数：STC-Xmin：（章节 3.1.2 中的方法，本文提出）、STC-TF：（章节 3.1.1 中的第三种方法）。纵向是嵌入率，表格里的值是 ODG。

表 3.2 不同负载和不同 cost_func 下的 ODG

隐写容量（bpc）	失真函数	
	STC-Xmin（proposed）	STC-Tf[11]
1/8	-1.8595	-0.9214
1/16	-1.1585	-0.6677
1/32	-0.8031	-0.5520
1/64	-0.6400	-0.5086
1/128	-0.5717	-0.5062
1/256	-0.5424	-0.5081
1/512	-0.5289	-0.5105

根据表格中的列数据，可以观察到，负载越低，ODG 绝对值越小（向 0 逼近，音频失真越小），负载越高，ODG 绝对值越大（向-4 逼近，音频失真越大）。两列 ODG 数值大致符合规律。对于失真函数 Tf，在负载低于 1/64 的时候。随着负载的降低，ODG 数值趋于平缓甚至有绝对值增大的趋势。

根据表格中的行数据：在列举的负载情况下，Tf 表现都比 Xmin 好。可以观察到趋势：在 1/512 甚至更低嵌入率下，Xmin 和 Tf 方法效果相当。

综上，本方案在不可感知性方面表现良好。

3.3.3 不可检测性

本实验使用的770段WAV样本均来自互联网公开样本，其采样率是48 kHz，位深度 32 bit，双声道，时长 5s。它们用于构建三个不同的 AAC 样本库，包括一个 cover 样本库（CDB）以及两个 stego 样本库（SDB_1, SDB_2）：

CDB：将 WAV 音频用公开的 AAC 音频编解码器 FAAC 进行编码，将最终得到的 AAC 文件作为 cover 音频样本，共 770 段。

SDB_1：对于 770 段音频样本，采用 Xmin 失真函数，按照 1/8、1/16、1/32、1/64、1/128、1/256、1/512 的负载进行 STC 嵌入，分别得到相应的 AAC 文件，即 770×7 段 stego 样本。

SDB_2：对于 770 段音频样本，采用 Tf 失真函数，按照 1/8、1/16、1/32、1/64、1/128、1/256、1/512 的负载进行 STC 嵌入，分别得到相应的 AAC 文件，即 770×7 段 stego 样本。

实验中，假设隐写分析者已知 STC 隐写中的失真函数，未知负载。基于此，将训练集（TRN）和测试集（TST）分配如下：

TRN：从 CDB 中选择 77×7 段 cover。相应地，在 SDB_1 或者 SDB_2 中，从每一种负载分别选择 77 段，即一共选择的是 77×7 段 stego。

TST：从 CDB 中选择 77×3 段 cover。相应地，在 SDB_1 或者 SDB_2 中，从某一种待测试的负载下选择 77×3 段 stego。

训练集和测试集准备就绪，隐写分析工作启动，包括特征提取和分类两个步骤。1）特征提取，即从 cover 和 stego 集中提取隐写分析特征。提取的隐写分析特征[18]有三个因素：关系类型（Markov 转移概率、累积邻近节点密度）、微分矩阵的阶（一阶、二阶）、特征构造的方向（帧间、帧内）。将 Qmdct 系数的范围调整在[-4,4]，所以每个音频样本特征的维度（dimensionality）是 $9\times9\times(2\text{^}3)=648$。2）分类（包括训练、测试过程），即把提取出来的特征放入隐写分析器，以此来分类 cover 和 stego。鉴于集成分类器[19]（Ensemble classifier）在高维特征上具备良好性能，这里采用它作为隐写分析器。

设定载密为正例，载体为反例。隐写分析实验结果用检测率 P_D 来衡量。检测率采用真正例率（True Positive Rate，TPR）和真反例率（True Negative Rate ，TNR）来表示，具体计算方法如下式（3.4-3.6）所示：

$$TPR = \frac{\text{载密音频被检测为载密音频的个数}}{\text{载密音频的总数}} \quad \text{（式 3.4）}$$

$$TNR = \frac{\text{载体音频被检测为载体音频的个数}}{\text{载体音频的总数}} \quad \text{（式 3.5）}$$

$$P_D = 1 - \frac{P_{MD}+P_{FA}}{2} = \frac{TPR+TNR}{2} \quad \text{（式 3.6）}$$

根据训练集（TRN）和测试集（TST）中的 cover-stego 对（这里 cover 就是由原始 WAV 压缩得到的 AAC 音频，stego 是不同负载下使用不同失真函数嵌入得到的载密 AAC 音频），得到的检测率如表 3.3。

表 3.3 不同负载和不同 cost_func 下的检测率

隐写容量（bpc）	失真函数	
	STC-Xmin（proposed）	STC-Tf[11]
1/8	0.8597 (+/- 0.0316)	0.6775 (+/- 0.0202)
1/16	0.8357 (+/- 0.0256)	0.9017 (+/- 0.0080)
1/32	0.9494 (+/- 0.0034)	0.8978 (+/- 0.0055)
1/64	0.9173 (+/- 0.0090)	0.7786 (+/- 0.0056)
1/128	0.7416 (+/- 0.0066)	0.6225 (+/- 0.0084)
1/256	0.5952 (+/- 0.0074)	0.5576 (+/- 0.0058)
1/512	0.5327 (+/- 0.0057)	0.5214 (+/- 0.0051)

由表格中的列数据可知，对于失真函数 Xmin，在 1/32、1/64 两种负载下的检测率较高。对于失真函数 Tf，在 1/16、1/32 两种负载下的检测率较高。其他负载下，随着负载的降低，检测率在降低。

由行数据可知，在负载 1/16 的情况下，本文提出的失真函数 Xmin 比 Tf 的检测率要低，即不可检测性更好。但是其他情况下，Tf 的表现都比 Xmin 要好。总体而言，本文提出的失真函数在不可检测性方面有待加强。

3.4 本章小结

本章根据最小化失真隐写理论和人耳听觉掩蔽效应，设计了基于心理声学模型的失真代价函数，解决了 STC 编码预备工作中的重要问题，实现了基于 STC 编码的 AAC 频域（即 AAC-Qmdct）的安全隐写算法。实验证明，本方案具备较大的隐写容量，可以达到 1/8 bpc；不可感知性方面表现良好，当隐写容量在 1/128 bpc 以及以下时，本方法的不可感知性与现有方法[11]相当。不足之处是不可检测性一般，有待改善。

第四章 音频隐写系统的设计与实现

本章基于 python tkinter[20]界面设计技术，集成了本文所提出的基于 STC 编码的 AAC-Qmdct 隐写算法，实现了一款音频隐写系统。隐写工作开始之前自然是要准备好载体文件和秘密信息文件，设定嵌入率（即隐写容量）和失真函数是应用 STC 隐写框架的预备工作，完整的隐写过程要求消息可以嵌入必定可以提取，基于以上考虑，本系统分别实现了相应的功能。

4.1 功能概览

基于第三章描述的基于 STC 编码的 AAC-Qmdct 隐写方案，本文实现了一款 AAC 音频隐写系统。系统模块功能包括：选择失真函数和嵌入率（隐写容量）、选择载体音频文件和秘密信息文件（过程含有隐写容量计算、调整秘密消息长度与隐写容量的关系）、消息嵌入和消息提取。本章从操作流程的层次上，将各个功能模块整理如下。

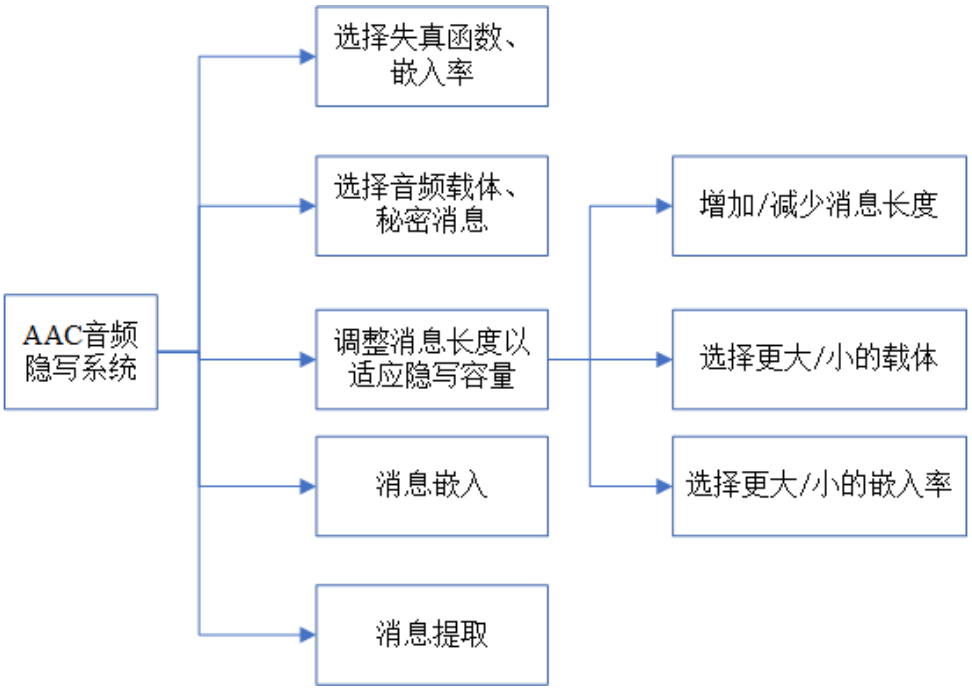


图 4.1 AAC 音频隐写系统-功能模块图

4.2 操作流程

4.2.1 启动概述

启动 AAC 音频隐写系统，系统弹出欢迎界面，如图 4.3 所示。点击“确定”，进入主界面，如图 4.2 所示。用户需要按照主界面上的控件（如按钮等）摆放顺序（从上到下，从左到右）进行操作：选择失真函数和嵌入率（隐写容量）、选择载体音频文件、选择秘密信息文件、开始嵌入、开始提取。

图 4.2　AAC 音频隐写系统-主界面

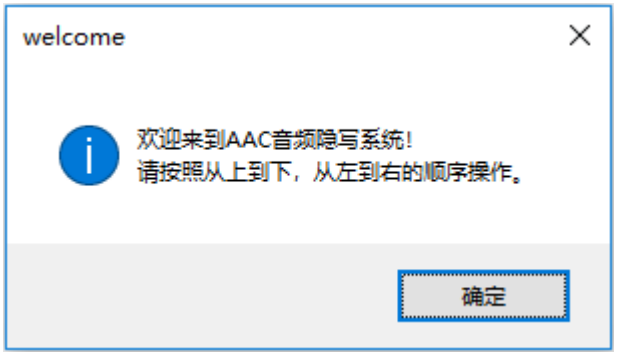


图 4.3　AAC 音频隐写系统-欢迎界面

4.2.2 选择失真函数和嵌入率

失真函数和嵌入率（隐写容量）的选择是应用 STC 编码框架的重要预备工作。用户操作流程如下：

（1）选择失真函数。备选项分别是 Xmin（3.1.2 章节中的方法）和 Tf 失真函数（3.1.1 章节中的第三种方法）。这里选择 Xmin。

（2）选择嵌入率（即 3.3.1 中描述的负载）。备选项有 7 个，分别是'1/8', '1/16', '1/32', '1/64', '1/128', '1/256', '1/512'。这里选择 1/64。

4.2.3 选择载体音频文件和秘密信息文件

用户选择载体音频的同时，系统计算隐写容量。在选择秘密信息文件的同时，系统将隐写容量（载体长度×嵌入率）与消息长度进行比对，提出一些备选建议。用户操作流程如下：

（1）点击“选择载体音频文件”，系统弹出文件对话框，如图 4.4。这里以“233.wav”为例。系统根据用户选择的载体长度和嵌入率计算隐写容量，并给出提示，如图 4.5。

（2）点击“选择秘密信息文件”，系统弹出文件对话框，如图 4.6。如果消息长度较短（隐写容量大于消息长度），系统给出解决方法提示框（可以增加秘密信息长度，直接执行下一步嵌入操作也可），如图 4.7。如果消息长度较长（隐写容量小于消息长度），系统给出解决方法提示框（减少秘密信息长度、选择更大的载体或者更大的嵌入率），如图 4.8。如果消息长度恰好符合要求（隐写容量等于消息长度），系统提示可以进行下一步的嵌入操作，如图 4.9。

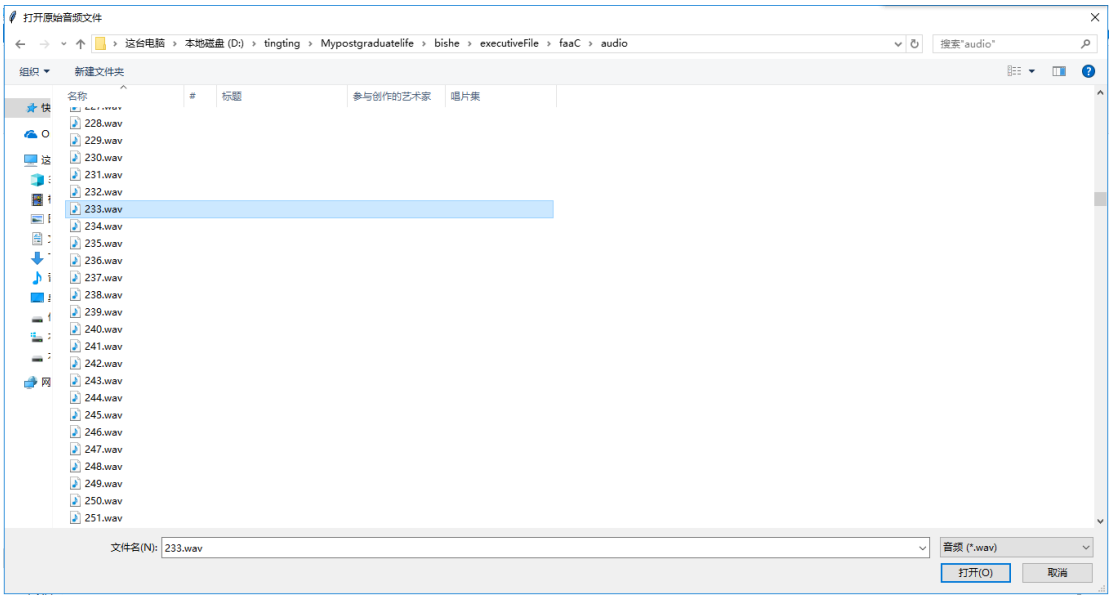


图 4.4 选择载体音频-文件对话框

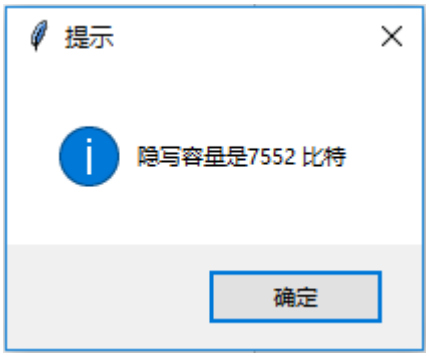


图 4.5 隐写容量提示框

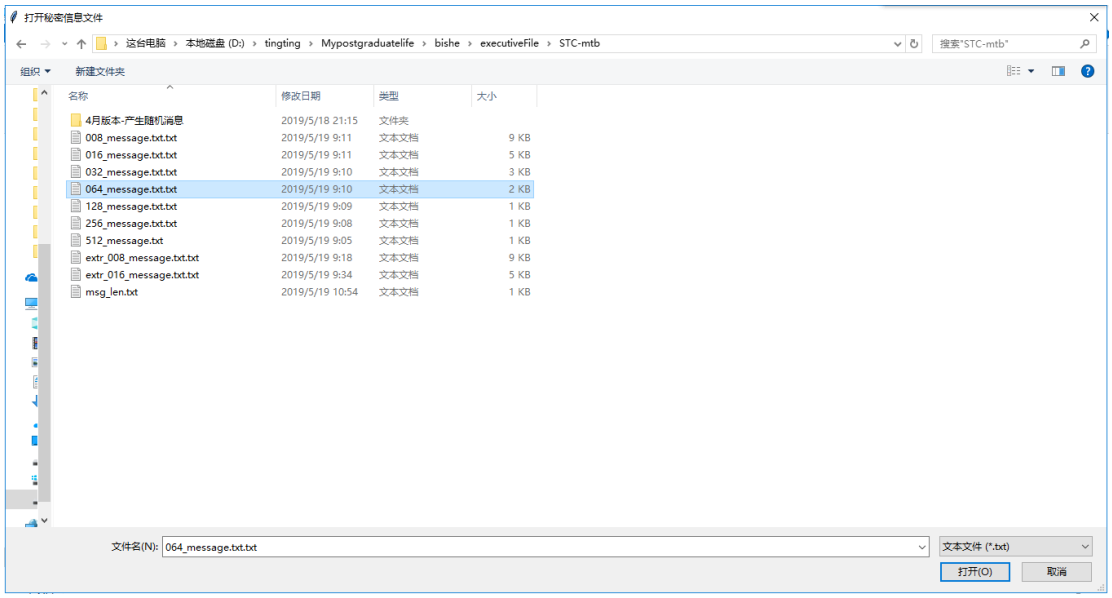


图 4.6 选择秘密信息-文件对话框

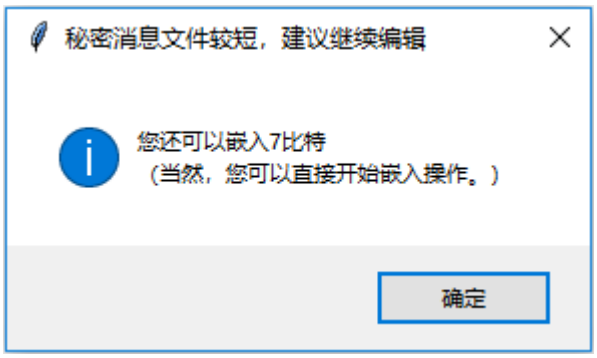


图 4.7 秘密信息长度较短-提示框

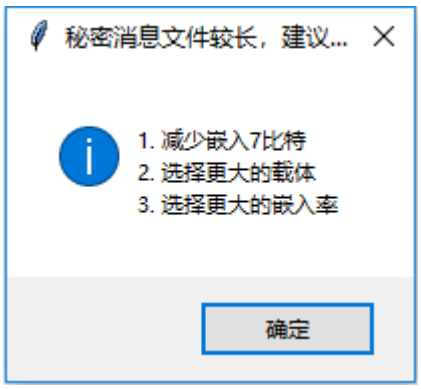


图 4.8 秘密信息长度较长-提示框

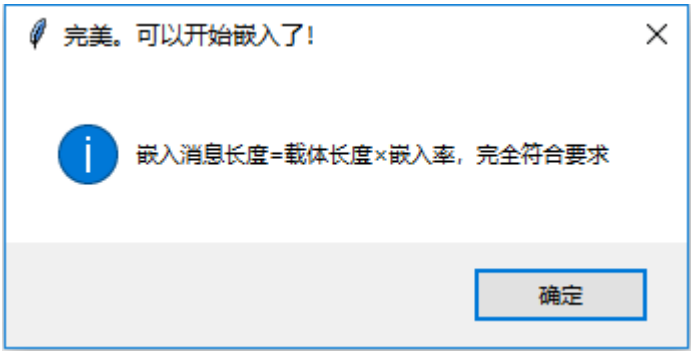


图 4.9 秘密信息长度符合要求-提示框

4.2.4 消息嵌入和消息提取

准备工作就绪，消息嵌入和消息提取工作开始。用户操作流程如下：

（1）点击“开始嵌入”，系统根据用户选择的失真函数、载体文件的 Qmdct、满足嵌入率要求的秘密消息，依据系统集成的基于 STC 编码的 AAC-Qmdct 算法，开始执行嵌入操作。嵌入完毕后，系统给出成功提示（如图 4.10），得到 233-Xmin-064.aac（以 Xmin 为失真函数，1/64 为嵌入率嵌入得到的，如图 4.11）。播放此载密音频，人耳无法听出与 233.wav（载体音频，如图 4.4）的差别。

（2）点击“开始提取”，系统根据消息长度、载密文件的 Qmdct，依据集成的算法执行提取操作。提取完毕后，系统给出成功提示，如图 4.12。根据提取出来的消息（如图 4.13）和嵌入之前选择的原始消息文件（如图 4.14）可知，两者内容一致，系统的嵌入和提取过程是完整且正确的。

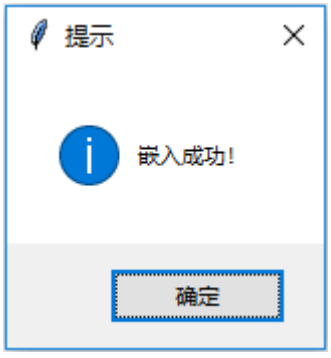
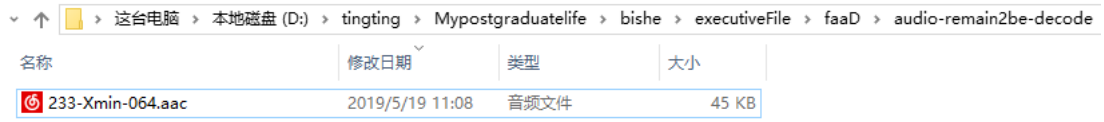
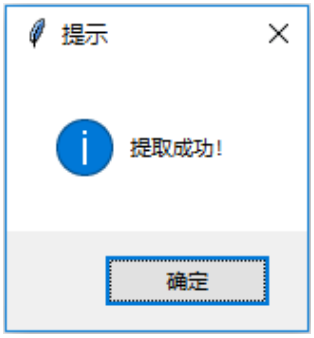


图 4.10 嵌入成功-提示框



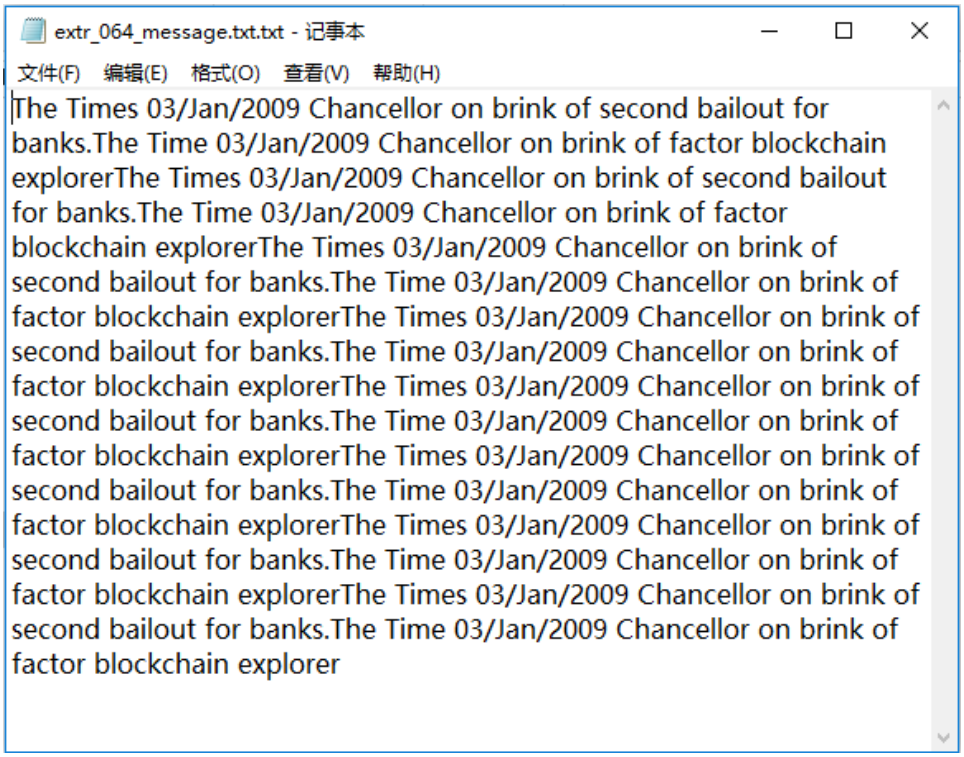
名称	修改日期	类型	大小
233-Xmin-064.aac	2019/5/19 11:08	音频文件	45 KB

图 4.11 载密 AAC 音频-文件资源管理器视图



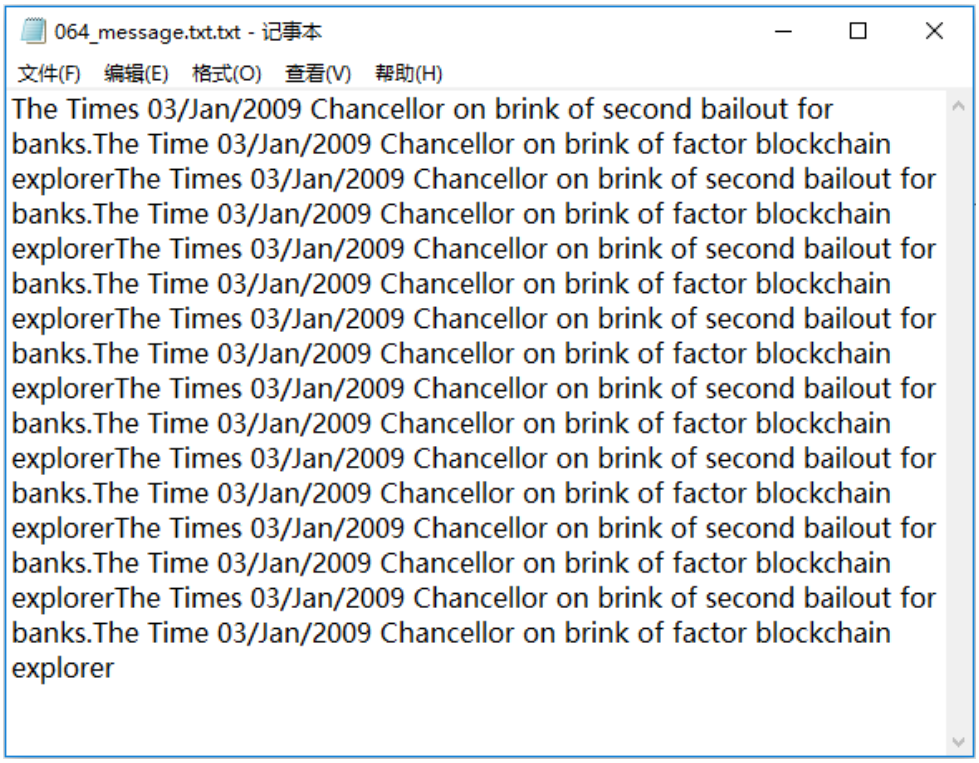
提取成功！

图 4.12 提取成功-提示框



The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorer

图 4.13 提取出的秘密信息-文本打开视图



The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorerThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks.The Time 03/Jan/2009 Chancellor on brink of factor blockchain explorer

图 4.14 原始的秘密信息-文本打开视图

4.3 本章小结

本章完成了一款 AAC 音频隐写系统，基于 python tkinter 界面设计技术，集成了本文所提出的基于 STC 编码的 AAC-Qmdct 隐写算法，实现了失真函数设置、嵌入率设置、音频载体导入、秘密消息文件导入、消息嵌入、消息提取的功能。此外，系统在设计之时为了增强鲁棒性，在音频载体导入、秘密消息文件导入模块增加了调整隐写容量与消息长度关系的解决方法。本章对系统的所有功能模块进行了操作流程的说明，结果证明界面交互流畅，用户体验较佳。

第五章 结论

5.1 本文总结

高级音频编码（AAC，Advanced Audio Coding）作为一种优秀的音频压缩标准已经得到了广泛使用，如 iTunes、YouTube、腾讯音乐等。AAC 音频的推广，为隐写研究带来了丰富的载体空间。为了达到隐写术的安全目标（不可感知性、不可检测性），设计面向 AAC 音频的安全隐写算法，本文的工作内容如下：

1）探究 AAC 编解码原理（尤其是量化模块、心理声学模型模块）和 STC 编码（STC，Syndrome-Trellis Codes），为设计失真代价函数、实现基于 STC 编码的 AAC-Qmdct 隐写算法奠定理论基础。

2)学习 PEAQ 工具的使用(计算隐写前后音频音质变化,测量不可感知性)、掌握与隐写分析相关的机器学习知识工具的使用（测量不可检测性），为实验评估工作做好技术储备。

3）根据 AAC 编码中的心理声学模型设计失真代价函数，实现了基于最小化失真隐写理论（STC 编码）的 AAC-Qmdct 隐写方案。实验结果表明，本方案在隐写容量、不可感知性方面表现良好，不可检测性有待提高。

4）实现了一款 AAC 音频隐写系统。本系统采用 python tkinter 界面技术，集成了本文提出的隐写方案，实现了文件导入、参数设置、消息嵌入和提取等功能。

通过探讨研究 AAC 编解码、STC 编码、有关隐写分析的知识以及工具，笔者对理论知识背景的理解得到了加深。本文根据 AAC 编码中的心理声学模型设计了失真函数，实现了基于 STC 编码的 AAC-Qmdct 隐写算法，笔者的思维表达能力得到了锻炼。在实验评估过程、完成音频隐写系统的过程中，笔者的动手实操能力得到了加强。本次研究工作具备应用价值，为互联网上的隐蔽通信提供了有效的技术解决方案。

5.2 未来展望

本文对于 AAC 音频量化 MDCT 系数的隐写工作有了一定的研究进展，但仍然存在很多不足之处。未来的工作方向打算从以下几个方面进行：

（1）本文实验证明，基于 STC 的隐写方法不可感知性良好，但是在不可检测性方面表现欠妥。由于现有的隐写方法对音频帧的相关性造成了破坏，后续在设计失真函数时，要将帧内帧间相关性的因素考虑进来。

（2）实验评估中，a)自变量使用的是负载，这个概念不够直观。未来需要将这个概念和相对嵌入率（嵌入信息码率/载体 AAC 文件的码率）联系起来（载体 AAC 文件的码率需要统一）。此外，b)音频样本采用的均为 48 kHz 采样率，不具备普适性，未来需要换成 44.1 kHz 的样本。

（3）在工程实现上，完成 1000 个音频的隐写耗费时长约 5 小时，实验需要的时间过长，有待优化。

致 谢

四年的本科生活即将完结，我满怀感恩，将本文写到了尾声。值此毕业论文完成之际，我将郑重地向所有关心和帮助过我的人们表示最真诚的谢意。

首先我想感谢我的导师。老师治学细心严谨，因材施教，每周都会进行工作汇报、组会讨论。有问题及时反馈，讨论解决方案，调整工作方向。非常感谢每周的汇报和交流，这点对我有很大的帮助。通过这些我懂得了做科研的过程：前期充分储备好理论知识、调研研究现状；后期动手实践保持 research 精神，不惧挑战，突破瓶颈。

本科四年里设置的专业课课程让我夯实了理论基础，在此衷心感谢所有专业课的任课老师。得益于老师的教诲，我明白了很多时候我们不能拘泥于本领域的知识，只有扩展知识面涉猎其他研究领域，才能获得灵感来源，开拓创新。

此外，感谢大学以来所有帮助过我的同学们。你们对我提出的学习建议都是宝贵的方法论，你们的陪伴是我的精神壁垒。

家是心灵栖息之地。感谢爸爸妈妈的支持和鼓励，你们给予了我前进的动力。常言道有国才有家，我们青年人应当立志报效祖国，将个人理想同国家命运结合，不负社会对人才的期待。

参考文献

[1] 杨榆, 雷敏. 信息隐藏与数字水印[M]. 北京: 北京邮电大学出版社, 2017.

[2] Yan S, Tang G, Chen Y. Incorporating data hiding into G.729 speech codec[J]. Multimedia tools and applications, 2016, 75(18): 11493-11512.

[3] AudioCoding.com. FAAC [EB/OL]. http://www.petitcolas.net/fabien/steganography/mp3stego,2006

[4] 魏一方. 音频中的信息隐藏和隐藏分析技术研究[D]. 中国科学技术大学, 2011.

[5] Zhu J, Wang R, Yan D. The Sign Bits of Huffman Codeword-Based Steganography for AAC Audio[C]. 2010 International Conference on Multimedia Technology, 2010: 1-4.

[6] 王昱洁, 郭立, 王翠平. 一种以 AAC 压缩音频为载体的隐写方法[J]. 小型微型计算机系统, 2011, 32(7): 1465-1468.

[7] Pinel J, Girin L, Baras C, et al. A high-capacity watermarking technique for audio signals based on MDCT-domain quantization[C]. Int. Congress on Acoustics, 2010.

[8] Filler T, Judas J, Fridrich J. Minimizing Embedding Impact in Steganography using Trellis-Coded Quantization[C]. Media forensics and security II, 2010: 754105:1-754105:14.

[9] Luo W, Zhang Y, Li H. Adaptive Audio Steganography Based on Advanced Audio Coding and Syndrome-Trellis Coding[C], 2017: 177-186.

[10] 张悦, 骆伟祺. 一种基于压缩域的 MP3 自适应隐写方法: CN106228981A[P]. 2016-08-03.

[11] Yang K, Yi X, Zhao X, et al. Adaptive MP3 Steganography Using Equal Length Entropy Codes Substitution[C], 2017: 202-216.

[12] M. B, R.E. G. Introduction to Digital Audio Coding and Standards[M]. The Springer International Series in Engineering and Computer Science,vol 721. Springer, Boston, MA, 2003.

[13] Iso/Iec. 13818-7,Information technology — Generic coding of moving pictures and associated audio information — Part 7: Advanced Audio Coding (AAC)[S], 1997.

[14] Filler T, Judas J, Fridrich J. Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes[J]. Ieee Transactions on Information Forensics and Security, 2011, 6(3): 920-935.

[15] Barni M. Steganography in Digital Media: Principles, Algorithms, and Applications (Fridrich, J. 2010) [Book Reviews][J]. IEEE Signal Processing Magazine, 2011, 28(5): 142-144.

[16] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[17] Itu-R. BS.1387-1,Method for objective measurements of perceived audio quality[S], 2001.

[18] Ren Y, Xiong Q, Wang L: A Steganalysis Scheme for AAC Audio Based on MDCT Difference Between Intra and Inter Frame, Kraetzer C, Shi Y Q, Dittmann J, Kim H J, editor, Digital Forensics and Watermarking, 2017: 217-231.

[19] Kodovsky J, Fridrich J, Holub V. Ensemble Classifiers for Steganalysis of Digital Media[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 432-444.

[20] David Beazley, Jones B K. Python Cookbook(3rd Edition)[M]. O'Reilly Media, Inc, 2013.