



一种针对 UnderMP3Cover 的 RS 隐写分析新方法

张坚, 王让定, 严迪群

(宁波大学信息科学与工程学院, 浙江 宁波 315211)

摘要: 提出了一种针对 UnderMP3Cover 的 RS (regular-singular) 隐写分析新算法。通过分析 RS 算法产生误差的原因, 选取了更适合的分辨函数, 修正了不同码率下的 MP3 初始偏差, 并改进了拟合方式, 同时, 在实验对比中选择了效果更佳的翻转算子。实验结果表明, 算法可以有效地估计出秘密信息的嵌入容量, 并将平均误差降低到 2%, 优于现有算法。

关键词: 隐写分析; UnderMP3Cover 隐写算法; RS 隐写分析方法

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2018044

A new RS steganalysis algorithm of detecting UnderMP3Cover

ZHANG Jian, WANG Rangding, YAN Diquan

College of Information Science and Engineering, Ningbo University, Ningbo 315211, China

Abstract: A new RS steganalysis algorithm to attack UnderMP3Cover was proposed. By analyzing the error caused by the RS algorithm, the choice of resolution function, the initial bias and the fitting method were modified, the flip operator was also selected in the experiment. Experimental results demonstrate that the proposed method can estimate the length of embedded message effectively and the average error reduces to 2%, which is better than the existing algorithms.

Key words: steganalysis, UnderMP3Cover, RS

1 引言

随着互联网技术的飞速发展, 多媒体技术的应用日益广泛^[1,2]。音频、图像和视频等多媒体在内容上存在很大的感官冗余, 人们可以利用这些

冗余嵌入秘密信息, 进行隐蔽通信, 即“隐写”。隐写分析作为检测隐写行为的技术, 日渐成为国内外研究关注的热点^[3]。

在多媒体信息中, 音频的发展较为迅速。MP3 (MPEG audio layer 3) 作为一种具备高压缩率和高

收稿日期: 2017-09-06; 修回日期: 2017-10-26

通信作者: 王让定, wangrangding@nbu.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61672302, No.U1736215); 浙江省自然科学基金资助项目 (No.LZ15F020002, No.LY17F020010); 宁波市自然科学基金资助项目 (No.2017A610123); 宁波大学科研基金资助项目 (No.XKXL1509, No.XKXL1503); 宁波大学王宽诚幸福基金资助项目

Foundation Items: The National Natural Science Foundation of China (No. 61672302, No.U1736215), Natural Science Foundation of Zhejiang Province of China (No.LZ15F020002, No.LY17F020010), Ningbo Natural Science Foundation (No.2017A610123), The Scientific Research Foundation of Ningbo University (No.XKXL1509, No. XKXL1503), K C Wong Magna Fund in Ningbo University

音质的音频有损压缩编码标准,是当前流传最广的音频格式之一。MP3 音频编码过程复杂,隐写容量大,较适合隐写,因此,针对 MP3 音频隐写进行深入的隐写分析研究具有意义^[4]。针对 MP3 音频开发的隐写算法,目前主要有 MP3Stego^[5]、MP3Stegz^[6]和 UnderMP3Cover^[7]3 种。然而,对应的隐写分析方法,大部分都是针对 MP3Stego 这种工具提出的,针对 UnderMP3Cover 的隐写分析则涉及较少。

UnderMP3Cover 是通过 LSB 算法进行连续或不连续嵌入的隐写算法,而 Fridrich 等人^[8]提出的 RS (regular-singular) 算法是目前唯一能够检测连续 LSB 替换和不连续 LSB 替换嵌入的秘密信息,且能可靠估计嵌入秘密信息大小的算法^[9]。因此 Jin 等人^[10]引入 RS 算法,提出了一种针对 UnderMP3Cover 的 RS 算法,并根据音频的帧重叠特性,采用了重叠分组的方式。虽然该方法能有效地估计出 UnderMP3Cover 隐写的嵌入比例,但仍存在约 4% 的误差。另外,他们在实验中仅测试了 128 kbit/s 的 MP3 音频,并没有给出其他码率的 MP3 的检测结果。

近年来,在图像隐写分析中,已有一些基于 RS 算法的改进算法。如 Ker 在参考文献[11,12]中提出的 IRS 算法、Luo 等人^[13]提出的 DRS 算法以及丁晋晶等人^[14]提出的 QRS 算法。这些算法针对载体图像的初始偏差和 RS 算法的拟合方式进行了改进,提高了检测效率。但他们都是基于图像的 RS 曲线拟合的,不能直接应用在音频隐写分析上。参考文献[10]也指出,采用图像中 RS 算法的拟合方式计算 UnderMP3Cover 嵌入率,得出的结果会产生误差。

本文在分析了 RS 算法误差产生原理的基础上,根据音频特性,提出了一种新的针对 UnderMP3Cover 的隐写分析方法。该方案采用了更适合的分辨函数,修正了各码率 MP3 的初始偏差,改进了拟合方式,并在实验对比中,选

择了效果更佳的翻转算子。最后,给出了 4 种主流码率下,本文算法与参考文献[10]的检测性能对比。实验结果表明,在不同的隐写嵌入率下,本文算法可以更为准确地估计出秘密信息的长度,且秘密信息的估计值与实际值的平均偏差仅为 2% 左右。

2 UnderMP3Cover 隐写原理

UnderMP3Cover 是 Platt C^[7]提出的一种典型的开源 MP3 隐写工具。它通过修改颗粒边信息中的编码参数——全局增益(global_gain)来实现秘密信息的顺序嵌入,其具体嵌入方式是 LSB 替换,即直接用秘密比特替换参数 global_gain 的最低有效位。但在可嵌颗粒的选择上,它通过使用一个参数 BitSpacing 来决定可嵌的颗粒间隔。

UnderMP3Cover 隐写过程如图 1 所示,假设待隐藏秘密信息为 $M = (m_1, m_2, \dots, m_k)$; 大致可以分为以下 4 个部分。

(1) 将待嵌 MP3 音频进行部分解码,提取每帧边信息中的参数 global_gain,将其按顺序排列,可以得到数组 $G = (gg_1, gg_2, \dots, gg_n)$ 。

(2) 判断当前颗粒是否为可嵌颗粒。若当前待嵌 global_gain 为 gg_i , 秘密信息比特 m_j , 则依据式(1)判断:

$$j = (i - 1) / Bs \quad (1)$$

其中, Bs 为参数 BitSpacing。若式(1)成立,则执行嵌入操作(3); 否则,继续查看下一个 global_gain,即将 i 加 1,继续步骤(2)。

(3) 按式(2)执行嵌入操作,当 $i \leq n$ 或 $j \leq k$ 中任一成立时执行步骤(4);

$$LSB(gg_{Bs \times i + 1}) \leftarrow (m_j) \quad (2)$$

其中, $LSB(x) \leftarrow (y)$ 为 LSB 替换操作,即 y 替换 x 的最低位。

(4) 将完成隐写的 global_gain 写回 MP3 边信

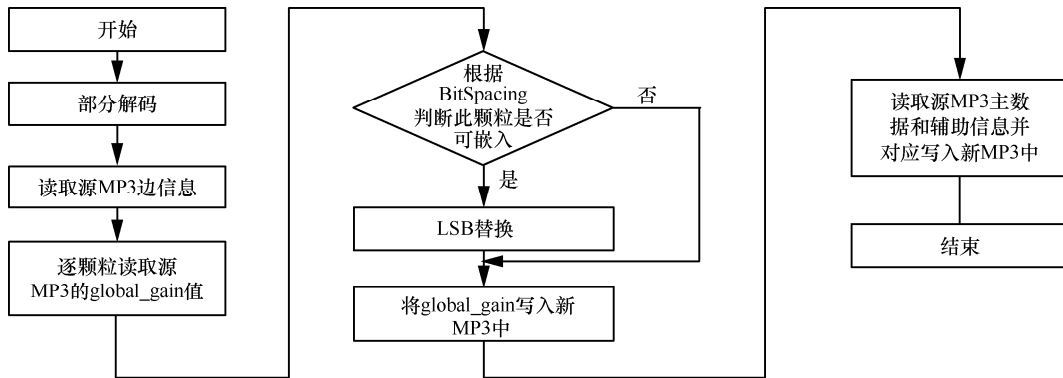


图1 UnderMP3Cover 隐写流程

息中，输出隐写后的 MP3 音频。

由于 UnderMP3Cover 存在特殊的嵌入控制参数 BitSpacing，且当 BitSpacing=1 时，UnderMP3Cover 的嵌入方式为顺序连续嵌入，而当 BitSpacing≠1 时，UnderMP3Cover 的嵌入方式即转变为顺序不连续嵌入，因而只能使用 RS 针对连续嵌入和不连续嵌入都有效的隐写检测方法。另外，参考文献[12]也已经通过实验证明了 RS 法是目前针对 LSB 隐写检测效果最好的方法之一。所以，借鉴 RS 方法，并将其改进，对 UnderMP3Cover 的隐写分析来说，是一种很好的思路。

3 RS 算法

3.1 RS 算法原理

Fridrich 在参考文献[8]中详细描述了 RS 算法的原理，简述如下。

对给定图像块的像素值以 Zigzag 的方式扫描，并排列成一个向量组 $G = (x_1, x_2, \dots, x_n)$ 。接着定义一个表征图像块空间相关性（平滑度和奇异性）的分辨函数 f （本文仿照参考文献[15]将定义相关性的函数称为分辨函数），其具体计算依据式（3）。 f 值越小，说明图像块相邻像素之间的起伏越小、空间相关性越强；反之，则表明图像块的空间相关性越弱。

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_i - x_{i+1}| \quad (3)$$

Fridrich 定义了一个翻转函数 F ：正翻转 F_1 、

0 翻转 F_0 、负翻转 F_{-1} ，翻转规则如式（4）。

$$\begin{aligned} F_1: 0 &\leftrightarrow 1, 2 \leftrightarrow 3, \dots, 208 \leftrightarrow 209 \\ F_0: F_0(x) &= x \\ F_{-1}: -1 &\leftrightarrow 0, 1 \leftrightarrow 2, \dots, 209 \leftrightarrow 210 \end{aligned} \quad (4)$$

通过翻转函数和翻转算子 $M = (M_1, M_2, M_3, M_4)$ 置换向量组内的值，其中 M_1, M_2, M_3, M_4 可取 0、1，置换式如式（5）。根据置换前后分辨函数 f 的大小关系，将像素组分为 R 组（ $f(F(G)) > f(G)$ ）和 S 组（ $f(F(G)) < f(G)$ ）。

$$F(G_i) = (F_{M_1}(x_1), F_{M_2}(x_2), F_{M_3}(x_3), F_{M_4}(x_4)) \quad (5)$$

利用 $M = (M_1, M_2, M_3, M_4)$ 置换所有像素组后，得到 R 和 S 组的百分比 $R_M\left(\frac{P}{2}\right)$ 和 $S_M\left(\frac{P}{2}\right)$ ，再将所有像素组 LSB 位进行 F_1 翻转，并再次计算置换后的 R 和 S 组的百分比，得到 $R_M\left(1 - \frac{P}{2}\right)$ 和 $S_M\left(1 - \frac{P}{2}\right)$ 。同理，利用 $-M = (-M_1, -M_2, -M_3, -M_4)$ 置换，同样可以得到 $R_{-M}\left(\frac{P}{2}\right)$ 、 $S_{-M}\left(\frac{P}{2}\right)$ 、 $R_{-M}\left(1 - \frac{P}{2}\right)$ 和 $S_{-M}\left(1 - \frac{P}{2}\right)$ 。

基于以上定义，Fridrich 根据大量图像，拟合出 RS 曲线，如图 2 所示。 x 轴为 LSB 发生改变的像素的百分比， y 轴为相应的 R_{-M} 、 R_M 、 S_{-M} 、 S_M 的值。设隐藏信息的嵌入率为 P ，因为 LSB 嵌入序列为 0、1 等概率随机序列，所以，LSB 发生改变的像素值百分比为 $\frac{P}{2}$ 。

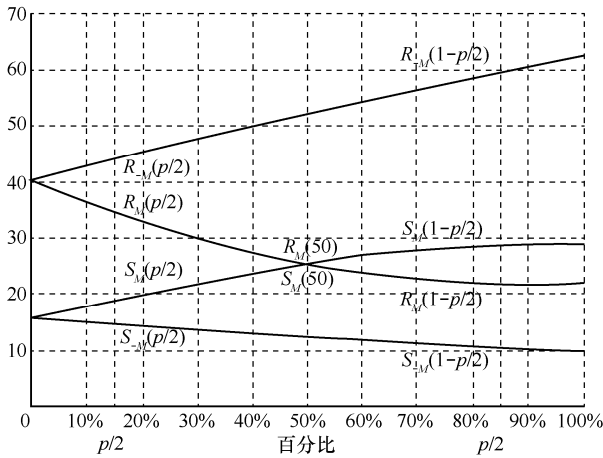


图2 Fridrich拟合的RS曲线

同时, Fridrich 根据大量的实验指出: R_{-M} 和 S_{-M} 与信息嵌入比率 P 近似呈线性关系, 而 R_M 和 S_M 曲线更接近二次曲线。且提出了两个假设。

当嵌入率为 0 时:

$$\begin{cases} R_M(0) = R_{-M}(0) \\ S_M(0) = S_{-M}(0) \end{cases} \quad (6)$$

当嵌入率为 100% 时:

$$R_M(0.5) = S_M(0.5) \quad (7)$$

基于式(6)和式(7)以及 $R_M\left(\frac{P}{2}\right)$ 、 $S_M\left(\frac{P}{2}\right)$ 、 $R_M\left(1-\frac{P}{2}\right)$ 、 $S_M\left(1-\frac{P}{2}\right)$ 、 $R_M\left(\frac{P}{2}\right)$ 、 $S_M\left(\frac{P}{2}\right)$ 、 $R_M\left(1-\frac{P}{2}\right)$ 和 $S_M\left(1-\frac{P}{2}\right)$ 这 8 个值进行拟合, 可建立一个求解嵌入率 P 值的二次方程, 参考文献[13]给出了详细的推导过程。

3.2 RS 算法的不足

由上述原理可知, RS 算法实质上是利用隐写前后相邻像素之间的相关性的变化, 实现隐写检测。因此, 分辨函数能否精确地表征像素间的相关性, 这是影响算法检测效率的关键因素。在原 RS 算法中, 表示相关性的分辨函数为式(3)。但这种分辨函数, 只考虑了相邻像素的相关性, 而实际上每一组像素之间均有相关性。所以, 为了

充分利用组内所有数据的相关性, 需要选取更佳的分辨函数。

Fridrich 在参考文献[8]中指出影响 RS 算法准确度的主要因素是载体图像的初始偏差。这是因为即使是自然载体, 由于随机的变化, 经过正负翻转, 分辨函数值也很大可能发生变化, 以至于得到长度非零的检测结果, 这一初始偏差会对检测算法的准确性有一定影响^[14]。表 1 给出了 4 000 首不同码率的原始音频的测试结果的平均值, 可以看出用 RS 方法测试, 即使是原始音频, 依然会估计出 4% 左右的隐写长度。

表 1 原始音频的检测结果的平均值

码率/(kbit·s ⁻¹)	64	128	192	256
估计长度	4.46%	3.89%	3.41%	4.75%

此外, 依据 RS 算法的原理, 本文使用 50 000 首第 5.1 节中的音频重新拟合了 RS 曲线, 如图 3 所示。可以明显看出, R_{-M} 和 R_M 、 S_{-M} 和 S_M 这 4 条曲线在 y 轴上均不相交。且 R_M 和 S_M 这两条曲线也没有在 $x=50$ 处相交, 即 RS 算法拟合过程中用到的两个假设(式(6)、式(7))只是近似成立。如果基于式(6)及式(7)来计算嵌入容量, 则会降低秘密信息长度的估计准确率。因此, 在拟合过程中需要采用新的拟合方式。

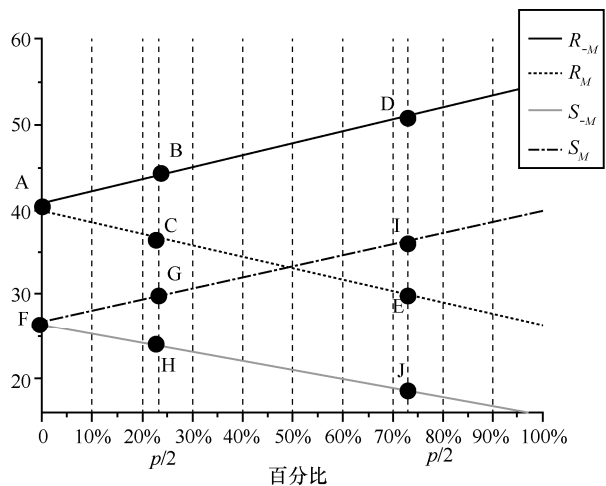


图3 本文拟合的RS曲线



4 本文算法

4.1 分辨函数的选取

为了充分利用组内所有数据的相关性,本文采用了概率论中的方差函数来描述组内数据的相关程度,即:

$$f(X_1, X_2, \dots, X_n) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (8)$$

方差函数不仅表达了样本偏离均值的程度,更能揭示样本内部彼此波动的程度。计算出的方差值越大,表明这组数据就越离散,相关性也就越强;相反方差越小,这组数据就越聚合,相关性也就越弱。第5节的实验也证明了方差函数可以提高RS算法的检测精度。

4.2 原始音频初始偏移量的修正

在RS算法下,绝大多数的原始载体图像都存在在或正或负的初始偏移量。在嵌入密度比较大的情况下,它对检测结果的影响相对较小,而当嵌入率小于或等于5%时,它的大小就与实际嵌入信息的长度相当了^[14]。

在理想情况下,若待测载体未经过LSB隐写,则翻转算子 M 和 $-M$ 会同等程度影响载体混乱程度^[10],即在RS曲线图中, R_{-M} 和 R_M 、 S_{-M} 和 S_M 均相交于 y 轴。而实际情况下,正负翻转算子的影响不可能等同,在图3中, R_{-M} 和 R_M 、 S_{-M} 和 S_M 曲线均未相交于 y 轴,交点偏出 y 轴的距离 x_0 ,在原RS算法拟合过程中被忽略了,因而产生了初始偏移量。同时注意到,不同码率下原始音频测试结果的方差很小,见表2。

表2 原始音频的检测结果的方差

码率/(kbit·s ⁻¹)	64	128	192	256
x_0	0.006 1	0.003 2	0.002 7	0.004 7

这说明不同的待测载体初始偏移量相差不大,进而说明引起初始偏移量的 x_0 值也相差不大。因此,本文借助Curve Fitting Tool,选取64 kbit/s、

128 kbit/s、192 kbit/s、256 kbit/s的不同嵌入率的音频,拟合出不同码率下的 x_0 值,以修正初始偏移量,修正值见表3。实验结果也证明对初始偏移量的修正,可提高检测精度,并用另一个库验证了拟合结果的适用性。

表3 各码率的修正值

码率/(kbit·s ⁻¹)	64	128	192	256
x_0	-0.544 4	-0.510 6	0.156 7	-1.067 6

4.3 拟合方式的改进

RS方法中,将 R_M 和 S_M 视为二次曲线,但在实际检测过程中,只能求出图3中C、E和G、I这4点,因此在推导过程中,需要借助 $R_M(0.5)$ 和 $S_M(0.5)$ 这两个点的值,并假设这两个值相等(即式(7))。但在第3.2节,已经分析过式(7)只是近似成立。

如果将 R_M 和 S_M 视为直线,那么在推导过程中就不需要借助式(7),但将二次曲线变为直线,拟合准确性下降,同样会引起误差。表4给出了在Curve Fitting Tool中,4条曲线分别用直线和二次曲线的拟合结果。

SSE是拟合误差的平方和,该值越接近0,说明拟合准确性越高。 R^2 是实际值与推测值之间相关系数的平方值。该值越接近1,则说明拟合准确性越高。综合这两个指标,可以看出采用二次曲线拟合的准确性略优于直线,但相差并不多。相比原算法中式(7)引起的误差,曲线变为直线引起的误差更小。之后的实验也证明这种拟合方式比原RS的方法更佳。

将4条曲线均视为直线后,可以得到两对相似三角形: $\triangle ABC \sim \triangle ADE$, $\triangle FGH \sim \triangle FIJ$ 。

可以得到:

$$\begin{cases} \frac{p_r/2 - x_0}{1 - p_r/2 - x_0} = \frac{R_{-M}(p/2) - R_M(p/2)}{R_{-M}(1 - p/2) - R_M(1 - p/2)} \\ \frac{p_s/2 - x_0}{1 - p_s/2 - x_0} = \frac{S_M(p/2) - S_{-M}(p/2)}{S_M(1 - p/2) - S_{-M}(1 - p/2)} \end{cases} \quad (9)$$

表4 RS 曲线的拟合结果

误差参数	拟合方式	R_{-M}	R_M	S_M	S_{-M}
SSE	直线	1.666×10^{-5}	1.682×10^{-5}	1.587×10^{-5}	1.055×10^{-5}
	二次曲线	1.6×10^{-5}	1.428×10^{-5}	9.187×10^{-6}	8.767×10^{-6}
R^2	直线	99.93%	99.94%	99.94%	99.91%
	二次曲线	99.93%	99.95%	99.96%	99.93%

其中, x_0 为表4中的修正值, $p_r/2 - x_0$ 为 $\triangle ABC$ 在 BC 边上的高, $p_s/2 - x_0$ 为 $\triangle FGH$ 在 GH 边上的高, 这两个值均可表征嵌入率的一半 $p/2$ 。进一步得到:

$$\begin{cases} p_r = 2 \cdot (d_1 - d_1 \cdot x_0 + d_2 \cdot x_0) / (d_1 + d_2) \\ p_s = 2 \cdot (d_3 - d_3 \cdot x_0 + d_4 \cdot x_0) / (d_3 + d_4) \end{cases} \quad (10)$$

其中:

$$\begin{aligned} d_1 &= R_{-M}\left(\frac{P}{2}\right) - R_M\left(\frac{P}{2}\right), d_2 = R_{-M}\left(1 - \frac{P}{2}\right) - R_M\left(1 - \frac{P}{2}\right) \\ d_3 &= S_M\left(\frac{P}{2}\right) - S_{-M}\left(\frac{P}{2}\right), d_4 = S_M\left(1 - \frac{P}{2}\right) - S_{-M}\left(1 - \frac{P}{2}\right) \end{aligned} \quad (11)$$

为了减小误差, 取了 R 曲线交点和 S 曲线交点横坐标的平均值, 得到:

$$p = \frac{p_r + p_s}{2} \quad (12)$$

4.4 算法的主要步骤

本文沿用参考文献[10]中重叠分组的方法, 并根据上述分析, 改进的 RS 算法可以分为以下步骤。

(1) 在解码端提取待测 MP3 音频每帧边信息中的全局增益参数 `global_gain`, 并将其按序存放一维数组 `gg[]` 中。

(2) 按顺序将 `gg[]` 分组, 要求每组系数个数相同, 但具体分组长度和分组滑动长度可任意设置。以分组长度取 6、滑动长度取 1 为例, 即 $G_i = (gg[i], gg[i+1], \dots, gg[i+5])$ 。

(3) 按式 (13) 计算各组元素之间的相关性 $f(G)$:

$$f(G) = \frac{1}{6} \sum_{n=i}^{i+5} (gg[n] - \overline{gg})^2, i \in [0, (gg.length - 4)] \quad (13)$$

(4) 对所有的组应用非负翻转, 如式 (13), 再计算翻转后每组的相关性 $f(F(G_i))$:

$$F(G_i) = (F_{M_1}(gg[i]), F_{M_2}(gg[i+1]), \dots, F_{M_6}(gg[i+5])) \quad (14)$$

其中, $(M_1, M_2, \dots, M_6)$ 为翻转类型算子, 其个数与分组长度相等, 每个元素的值可取 1、-1 或 0。因为这里使用的是非负翻转, 故 $M_1, M_2, \dots, M_6$ 只能取 0 或 1。

(5) 依据上述对正常组、异常组的定义, 统计这二者在所有组中的占比 $R_M\left(\frac{p}{2}\right)$ 和 $S_M\left(\frac{p}{2}\right)$,

其中 $M = (M_1, M_2, \dots, M_6)$ 为翻转类型算子的具体组合, p 为秘密信息的估计长度。

(6) 与步骤 (4) 和 (5) 同理, 对所有的组应用非正翻转, 并在计算翻转后每组相关性的基础上, 依据正常组、异常组的定义, 统计出 $R_{-M}\left(\frac{p}{2}\right)$ 和 $S_{-M}\left(\frac{p}{2}\right)$, 其中 $-M = -1 \times M$ 。

(7) 翻转 `gg` 数组中所有元素的 LSB 位, 并重复步骤 (5) ~ 步骤 (7) 得到 $R_M\left(1 - \frac{p}{2}\right)$, $S_M\left(1 - \frac{p}{2}\right)$, $R_{-M}\left(1 - \frac{p}{2}\right)$ 和 $S_{-M}\left(1 - \frac{p}{2}\right)$ 。

(8) 根据式 (10) 求出 p_r 和 p_s , 从而得到秘密信息的估计长度 $p = (p_r + p_s) / 2$ 。

5 实验结果和分析

5.1 实验设置

为了客观有效地评估本文方法的性能, 构造了一个基础音频数据库。该基础音频数据库的样本均是由 CD 抓轨得到的、44.1 kHz 采样率、16 bit



量化、单声道的 WAV 音频。这些 WAV 音频主要是从提供各种无损音乐下载的网站上下载获得。但通常下载的每首音频文件都是包含对应整张 CD 的全部内容。因此, 本文使用 Hifiwav、MATLAB 等能够实现音频裁剪功能而不对音频具体数据产生影响的软件对下载的 WAV 进行裁剪。最后, 将裁剪后的各种时长的音频样本收录到本文的音频数据库中。目前本数据库中裁剪后的音频样本共有 2 818 首, 时长均为 1 min。

从该数据库中随机选择 1 000 首的 WAV 音频, 用于测试本文所提方法。首先, 将这些 WAV 音频压缩成 4 种 (64 kbit/s、128 kbit/s、192 kbit/s 和 256 kbit/s) 不同码率的 MP3 音频, 作为未隐写的样本, 共 4 000 首。在此基础上, 使用 UnderMP3Cover 对这 4 000 首音频进行隐写, 对应生成 40 000 首不同隐写强度的隐写样本。需要说明的是, 被 UnderMP3Cover 隐写在音频样本中的秘密信息共有文本、音频和图像 3 种不同类型的文件, 但大小均为 10 byte。隐写时, 每首音频从以上 3 种文件中随机挑选一种进行嵌入。

每个样本时长 1 min, 包含 4 590 个颗粒的数据。因此当 BitSpacing=1 时理论满嵌容量为 4 560 bit, 但是如果 BitSpacing=2, 嵌入容量就

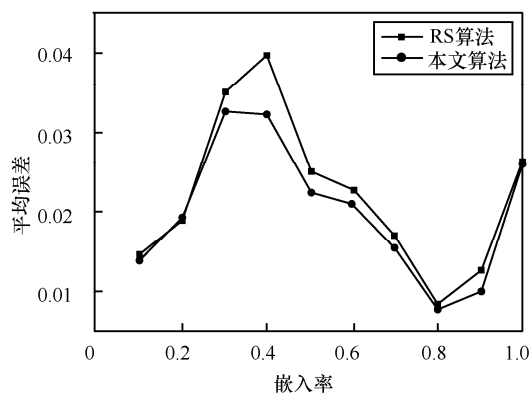
会折半, 同理, BitSpacing=3 时即为理论满嵌的 1/3。本实验中仅考虑 BitSpacing=1、2 和 3 的 3 种情况, 也即理论满嵌可以分别看成 100%、50%、33.3% 的情况。

使用 UnderMP3Cover 对每个样本的不同 BitSpacing 的情况分别制作出 10 种嵌入强度的隐写样本。BitSpacing=1 对应的是嵌入强度为 10%、20%、30%、...、100% 的样本, BitSpacing=2 对应的是嵌入强度为 5%、10%、15%、...、50% 的样本, 以此类推。

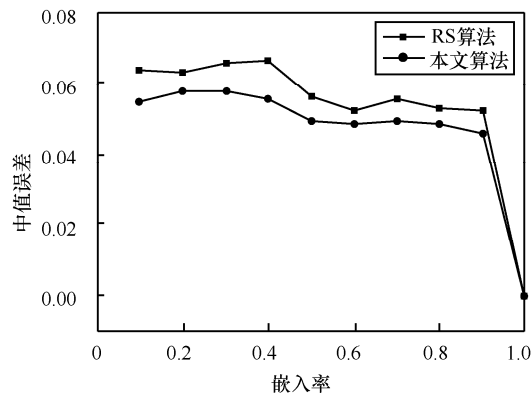
5.2 不同分辨函数的结果

RS 算法核心是隐写前后数据相关性的变化。因此分辨函数对相关性的描述准不准确, 对最后结果的精确性起着重要的作用。实验对比了 RS 算法的分辨函数和本文的方差函数对最终结果的影响。

图 4~图 6 分别是在参数 BitSpacing 为 1、2、3 的情况下, 不同分辨函数对秘密信息嵌入长度估计的平均误差 (所有测试样本估计值与实际值偏差的平均) 和中值误差 (所有测试样本估计值与实际值偏差的中位数)。可以看到, 在不同嵌入率下, 除了 BitSpacing=1 时, 平均误差优势不明显外, 其余情况下方差函数明显优于原 RS 算法中的分辨函数。



(a) 平均误差



(b) 中值误差

图 4 BitSpacing=1 时, 不同分辨函数的平均误差和中值误差

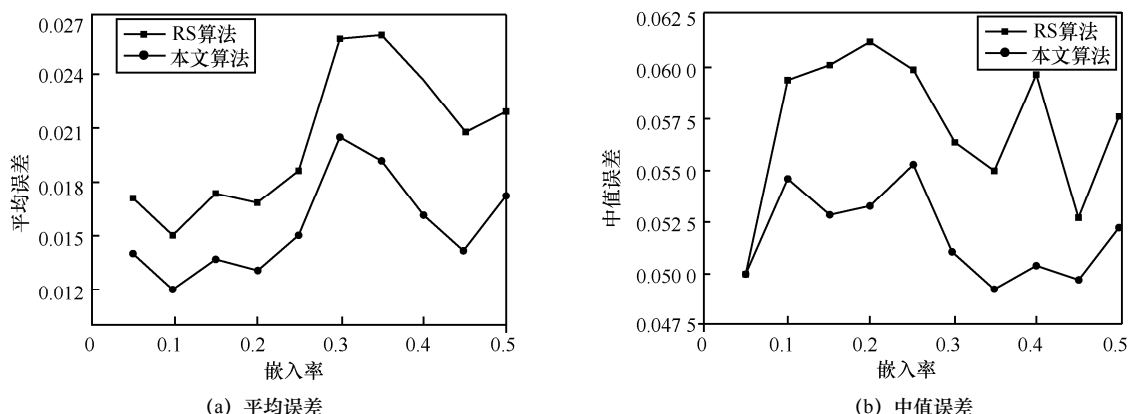


图5 BitSpacing=2时,不同分辨函数的平均误差和中值误差

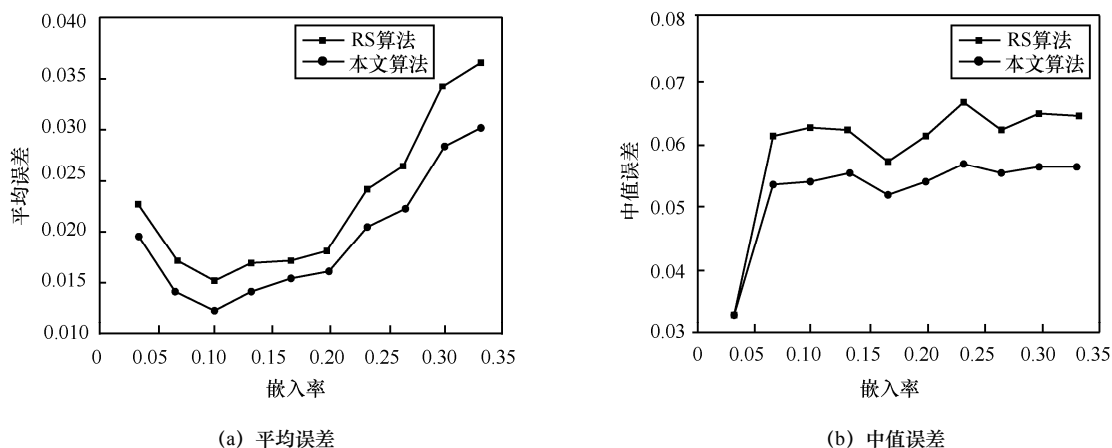


图6 BitSpacing=3时,不同分辨函数的平均误差和中值误差

5.3 不同翻转类型算子的结果

RS 分析法的检测性能一定程度上取决于翻转类型算子的选择,是因为翻转类型算子的具体组合决定了所加噪声的类型。Ker 等人在参考文献[11]中对常用的几种算子进行了研究,他们发现

(0110) 和 $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ 的翻转类型算子能使隐写分

析方法取得较好的性能。

尽管本节研究的对象是全局量化增益参数,且可供选择的翻转类型算子也必须是一维的,但选择怎样的算子也是本方法的关键之一。本文仿

照 Ker 等人在参考文献[11]中的工作,选了如下算子进行对比分析: (01)、(010)、(0110)、(01010)、(01110)、(010010)、(011110)、(0110110)、(0100010)、(0101010)。

图7~图9分别是在参数 BitSpacing 为 1、2 和 3 的情况下,10 种算子对秘密信息嵌入长度估计的误差值的平均值和中值。可以看到,在不同嵌入率下,(010010)在 BitSpacing=1 和 3 的情况下,基本都是平均误差最小的算子,只是在 BitSpacing=2 的情况下略差,且均优于原 RS 算法的算子(0110),而中值误差近似。因此综合以上 3 种情况,选取(010010)作为本方法的翻转算子。

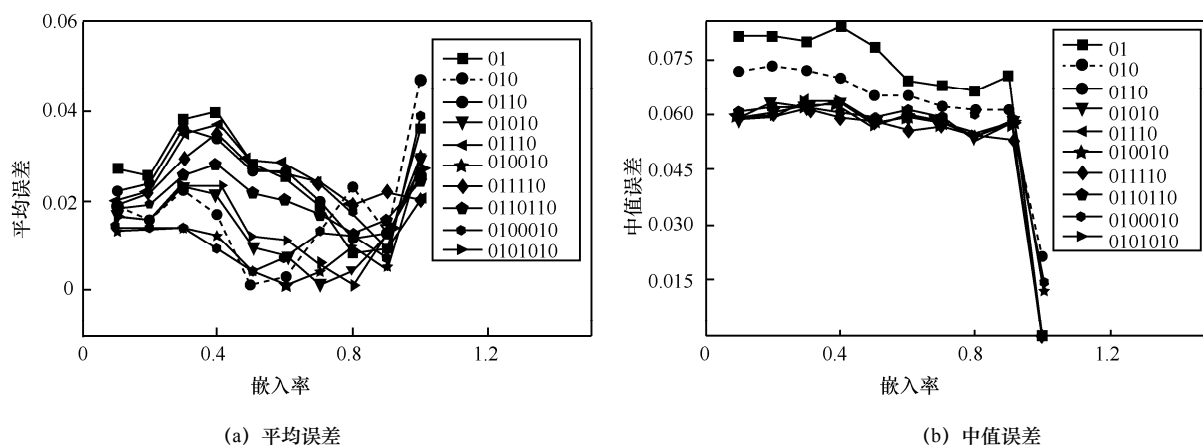


图 7 BitSpacing=1 时, 不同算子的平均误差和中值误差

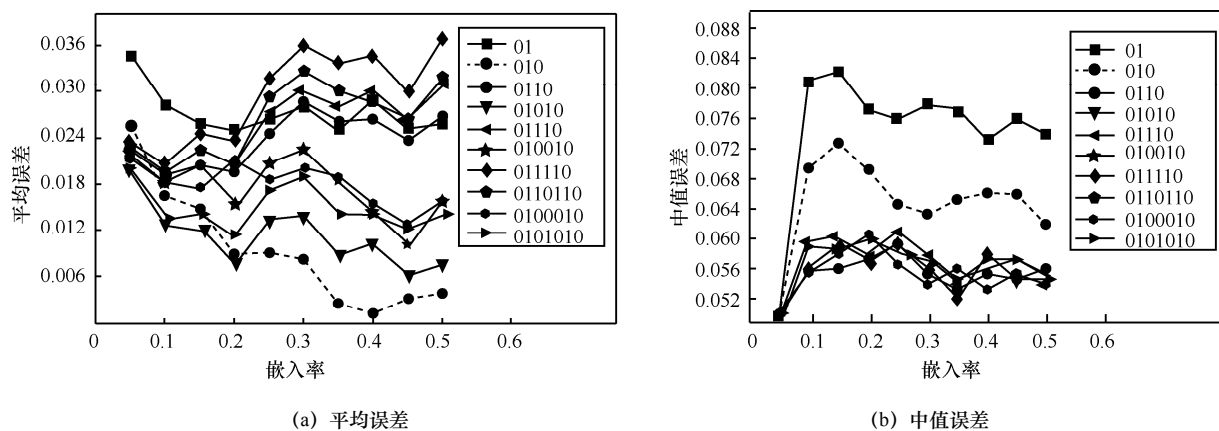


图 8 BitSpacing=2 时, 不同算子的平均误差和中值误差

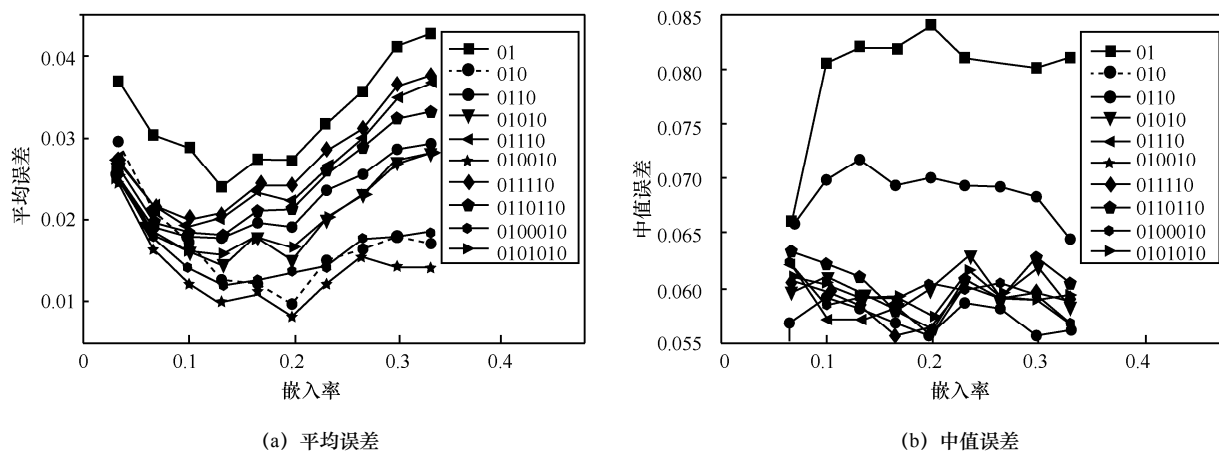


图 9 BitSpacing=3 时, 不同算子的平均误差和中值误差

5.4 不同拟合方式的结果

本算法摒弃了式 (7) 和式 (8), 对两个会引起较大误差的推论, 重新进行了拟合。图 10~

图 12 分别是在分辨函数为方差函数、算子为 (010010) 的情况下, 不同拟合方式的结果。在 BitSpacing 等于 1、2 和 3 的情况下, 本文算法拟

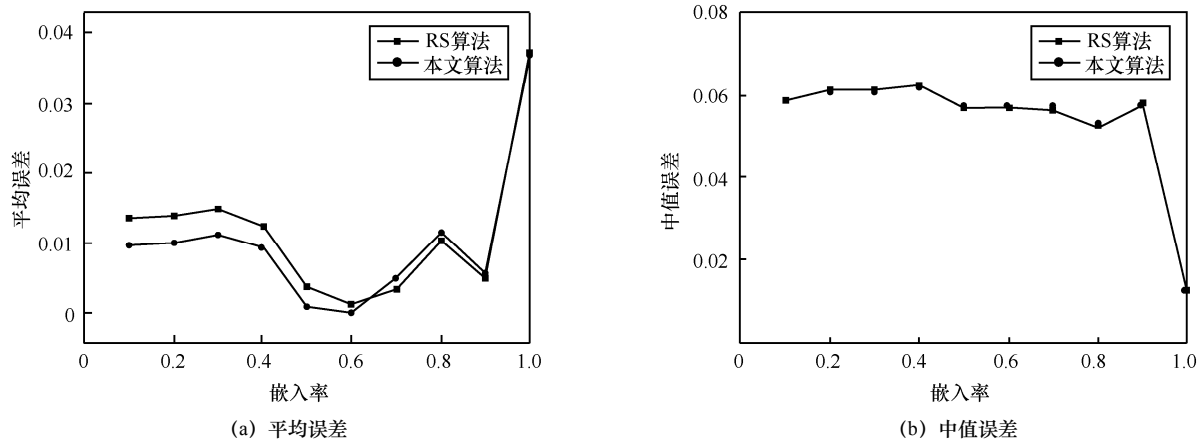


图 10 BitSpacing=1 时, 不同拟合方式的平均误差和中值误差

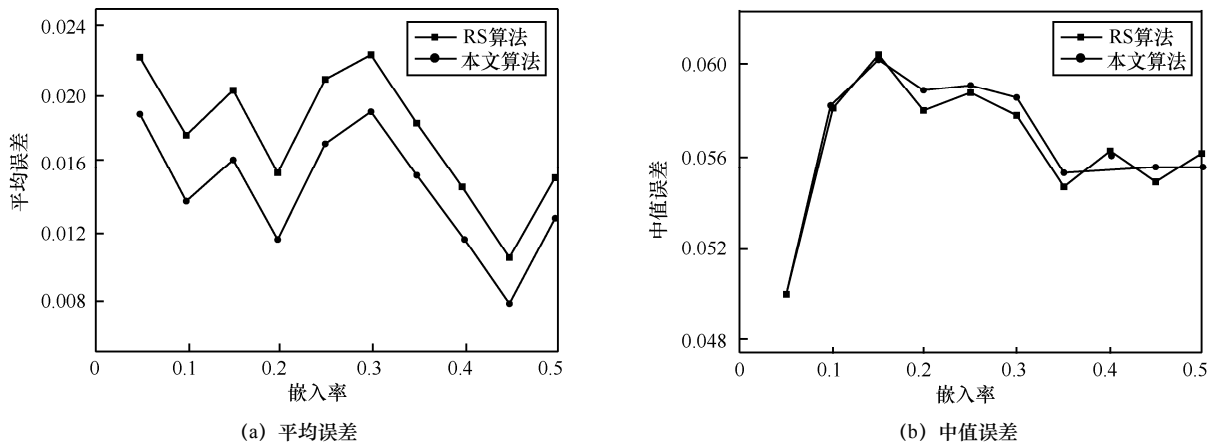


图 11 BitSpacing=2 时, 不同拟合方式的平均误差和中值误差

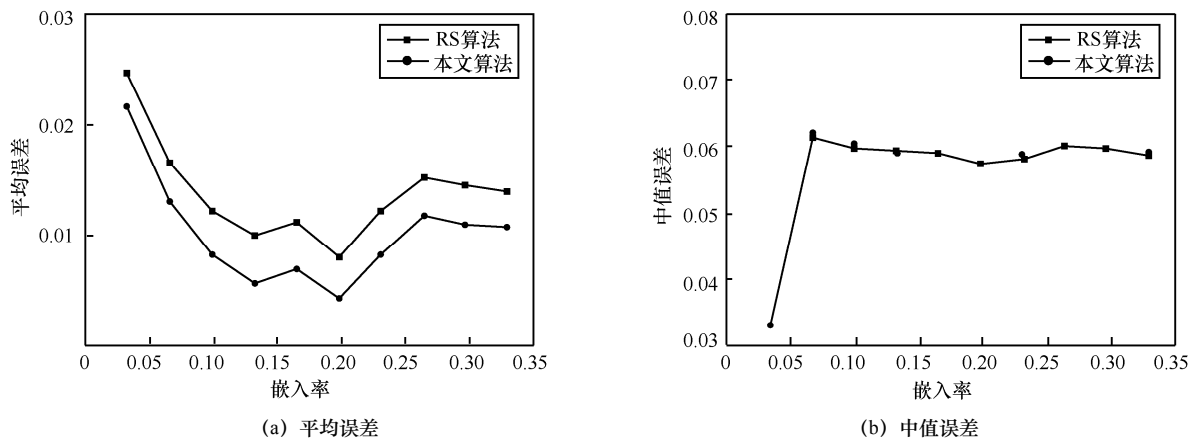


图 12 BitSpacing=3 时, 不同拟合方式的平均误差和中值误差

合方式的平均误差在 BitSpacing=2 和 3 时明显优于原 RS 的拟合方式, 在 BitSpacing=1 时略优。而 3 种情况下中值误差均相差不大。因此, 本文算法的拟合方式更佳。

5.5 修正值适用性的验证

本文是根据不同码率的音频, 用 Curve Fitting Tool 拟合出各码率对应的修正值, 但该值是否适用其他库的音频, 还需实验验证。本文另外选取



了 500 首 WAV 音频,按照第 5.1 节的定义,制作了同样的样本,结果以 128 kbit/s 为例。由表 5~表 7 可以看出,根据第 5.1 节的库拟合出的值,依然可以适用于其他库的音频,然而在低嵌入率下,略微差些,但总体而言,大部分嵌入率下,效果均优于未修正的结果,因此,本文拟合出的修正值具有一定的适用性。

5.6 不同码率的结果

目前,只有 Jin^[10]提出了针对 UnderMP3Cover 的隐写分析算法。因此,将本文算法与其进行对比。但在 Jin 的实验中,仅针对 128kbit/s 的音频进行了测试。事实上,MP3 音频码率越高,编码过程中量化就越精细,量化步长就越小,而全局增益变量 global_gain 就是量化步长的另一种表现

形式。因此码率不同,即使是同一个 WAV 音频,生成的 MP3 的 global_gain 也不同。表 8~表 10 是当 BitSpacing=1、2 和 3 的情况下,本文算法与 Jin 的算法在估计长度和中值误差的对比。可以看出,本文算法在估计长度方面,除了 BitSpacing=1 时,个别嵌入率差一点,其他均比 Jin 的结果更准确。而在 BitSpacing=1 和 2 时,平均误差大约为 1%,BitSpacing=3 时,平均误差约为 2%。相应的中值误差也略有提升。相比 Jin 的算法,本文算法可以更加精准地估算出秘密信息嵌入百分比。

6 结束语

本文针对 UnderMP3Cover 隐写算法,提出了一种改进的 RS 隐写分析算法。相比原 RS 方法,

表 5 BitSpacing=1 时,不同嵌入率的结果

拟合方式		嵌入率									
		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
估计长度	未修正	9.09	20.40	30.32	40.30	51.35	61.01	71.07	80.67	90.51	98.44
	修正	8.66	20.00	29.97	40.00	51.10	60.81	70.93	80.57	90.46	98.43
中值误差	未修正	3.33	3.02	3.16	3.06	3.27	2.79	2.81	2.98	2.62	0.07
	修正	3.44	2.93	3.18	3.19	3.37	2.75	2.73	2.91	2.64	0.06

表 6 BitSpacing=2 时,不同嵌入率的结果

拟合方式		嵌入率									
		5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
估计长度	未修正	4.22	9.35	15.02	20.53	26.54	32.21	37.65	43.19	48.43	52.80
	修正	3.84	8.91	14.59	20.13	26.17	31.87	37.33	42.91	48.17	52.57
中值误差	未修正	3.34	2.90	2.87	2.94	3.17	3.91	4.34	4.31	4.51	4.16
	修正	3.41	3.12	2.85	2.81	3.01	3.67	4.23	4.08	4.32	4.03

表 7 BitSpacing=3 时,不同嵌入率的结果

拟合方式		嵌入率									
		3.3%	6.6%	9.9%	13.2%	16.5%	19.8%	23.1%	26.4%	29.7%	33%
估计长度	未修正	2.66	5.74	7.82	11.21	14.80	18.32	21.84	25.29	28.91	34.66
	修正	2.13	4.69	9.38	12.85	16.61	20.23	24.09	27.52	31.21	32.38
中值误差	未修正	2.34	2.12	2.19	3.03	2.89	2.77	2.96	3.24	3.26	3.26
	修正	3.26	3.19	3.25	2.59	2.60	2.61	2.92	2.89	2.78	2.96

表8 BitSpacing=2 时, 各码率的音频不同嵌入率的结果

码率/(kbit·s ⁻¹)		检测 方法	嵌入率									
			10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
64	估计	Jin 算法	12.96	22.78	33.54	42.65	52.84	62.50	72.14	81.31	91.20	95.78
	长度	本文算法	11.63	21.04	31.08	40.39	50.31	60.21	69.89	79.26	88.20	94.82
	中值	Jin 算法	8.42	8.70	8.52	8.62	8.70	7.78	8.32	7.97	6.80	0.00
	误差	本文算法	7.75	7.88	7.51	7.85	7.77	7.53	7.68	7.94	6.96	2.09
128	估计	Jin 算法	12.30	22.48	33.67	43.69	52.64	62.39	71.60	80.74	91.28	97.23
	长度	本文算法	10.96	20.98	31.11	40.94	50.11	59.95	69.50	78.87	89.45	96.28
	中值	Jin 算法	6.44	6.63	6.67	6.58	5.98	5.88	5.65	5.50	5.76	0.00
	误差	本文算法	5.88	6.10	6.09	6.26	5.73	5.70	5.70	5.32	5.78	1.25
192	估计	Jin 算法	11.48	21.91	33.51	43.95	52.49	62.28	71.68	80.85	91.25	97.38
	长度	本文算法	10.81	20.98	31.41	41.55	50.35	60.26	69.88	79.25	89.83	96.69
	中值	Jin 算法	6.36	6.32	6.57	6.61	5.65	5.26	5.52	5.31	5.25	0.00
	误差	本文算法	5.43	5.67	5.47	5.34	5.14	5.05	5.23	5.23	4.87	0.49
256	估计	Jin 算法	13.29	23.73	34.89	44.05	53.38	62.82	71.87	80.97	90.93	97.16
	长度	本文算法	10.90	21.16	31.50	40.69	50.53	60.15	69.47	78.82	89.24	96.45
	中值	Jin 算法	6.78	7.14	6.86	6.83	6.56	6.25	5.84	5.86	5.59	0.00
	误差	本文算法	5.97	5.58	5.55	5.42	5.45	5.36	5.17	5.17	5.01	1.29

表9 BitSpacing=2 时, 各码率的音频不同嵌入率的结果

码率/(kbit·s ⁻¹)		检测 方法	嵌入率									
			5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
64	估计	Jin 算法	8.67	12.65	17.28	22.02	27.44	32.53	37.45	42.60	47.52	51.92
	长度	本文算法	7.78	11.96	16.57	21.12	26.26	31.28	36.24	40.78	45.82	50.29
	中值	Jin 算法	5.00	8.89	8.66	8.82	8.97	8.89	8.13	8.36	8.16	7.77
	误差	本文算法	5.00	8.05	8.31	7.77	8.00	7.85	7.38	7.48	7.66	7.04
128	估计	Jin 算法	7.51	12.16	17.22	22.25	27.67	33.01	37.71	42.85	47.63	52.75
	长度	本文算法	6.87	11.37	16.61	21.15	26.71	31.89	36.51	41.15	45.77	51.26
	中值	Jin 算法	5.00	6.56	6.40	6.28	5.92	6.24	6.10	5.97	6.22	6.16
	误差	本文算法	5.00	5.82	6.02	5.89	5.91	5.86	5.53	5.60	5.55	5.56
192	估计	Jin 算法	6.73	11.51	16.74	21.69	26.88	32.62	37.64	42.38	47.10	52.21
	长度	本文算法	6.61	11.33	16.46	21.15	26.33	31.87	36.68	41.09	45.77	50.95
	中值	Jin 算法	5.00	5.94	6.01	6.12	5.99	5.64	5.50	5.96	5.27	5.76
	误差	本文算法	5.00	5.77	5.56	5.51	5.66	5.09	4.93	5.25	4.86	5.26
256	估计	Jin 算法	8.53	13.15	18.44	23.29	28.22	33.78	38.60	43.39	48.09	53.35
	长度	本文算法	6.76	11.42	16.49	21.06	26.23	31.76	36.48	40.97	45.76	51.00
	中值	Jin 算法	5.00	6.61	6.69	6.58	6.60	6.69	6.24	6.32	6.39	6.24
	误差	本文算法	5.00	5.69	5.57	5.57	5.46	5.56	5.20	5.28	5.22	4.89

表10 BitSpacing=2 时, 各码率的音频不同嵌入率的结果

码率/(kbit·s ⁻¹)		检测 方法	嵌入率									
			3.3%	6.6%	9.9%	13.2%	16.5%	19.8%	23.1%	26.4%	29.7%	33.0%
64	估计	Jin 算法	7.42	9.82	13.00	16.03	18.77	21.96	25.49	29.12	31.86	35.02
	长度	本文算法	6.46	8.73	11.49	14.59	17.39	20.44	23.89	27.49	30.03	33.19
	中值	Jin 算法	3.30	6.60	8.80	8.83	8.11	8.89	8.12	8.62	8.12	8.23
	误差	本文算法	3.30	6.60	8.13	7.49	7.50	7.73	7.70	7.78	7.10	7.72
128	估计	Jin 算法	6.08	8.76	11.97	15.08	18.66	21.87	25.71	29.25	32.91	36.28
	长度	本文算法	5.45	7.90	10.73	13.77	17.20	20.21	23.93	27.57	30.79	34.07
	中值	Jin 算法	3.30	6.40	6.34	6.35	6.15	6.10	6.24	6.37	6.17	6.15
	误差	本文算法	3.30	6.24	6.01	5.91	5.90	5.74	5.87	6.01	5.96	5.91
192	估计	Jin 算法	5.61	8.34	11.44	14.92	18.25	21.64	25.56	29.09	33.19	36.74
	长度	本文算法	5.27	7.81	10.70	14.01	17.25	20.47	24.10	27.62	31.33	34.76
	中值	Jin 算法	3.30	6.13	6.26	6.20	5.71	6.11	6.66	6.21	6.47	6.45
	误差	本文算法	3.30	5.38	5.28	5.57	5.14	4.97	5.39	5.71	5.23	5.54
256	估计	Jin 算法	7.06	9.81	13.01	16.45	19.42	22.85	26.83	30.29	34.01	37.76
	长度	本文算法	5.34	7.80	10.74	14.00	16.86	20.11	23.93	27.29	30.86	34.53
	中值	Jin 算法	3.30	6.60	6.83	6.80	6.67	6.81	6.80	6.72	6.75	7.17
	误差	本文算法	3.30	5.58	5.58	5.94	5.76	5.51	5.62	5.46	5.55	5.65



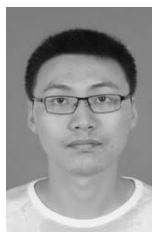
通过采用更合适的分辨函数和拟合方式,修正不同码率下MP3的初始偏差以及在实验中选取了更佳的翻转算子,提升了算法的精度,并将秘密信息的估计值与实际值的平均偏差降到了2%左右。此外,本文建立了包含4种主流码率的实验音频库,相比参考文献[10],给出了更为全面的结果。

虽然,本文算法较为准确地估计出UMC隐写音频中秘密信息嵌入的百分比,但秘密信息的嵌入方式是由参数BitSpacing控制的。因此,在不知BitSpacing值的情况下,仍然无法准确地提取出秘密信息。未来的工作中,将进一步研究如何估计BitSpacing的值,并最终实现对秘密信息的准确提取。

参考文献:

- [1] 郑志彬. 信息网络安全威胁及技术发展趋势[J]. 电信科学, 2009, 25(2): 28-34.
ZHENG Z B. Overview of mobile communication services security [J]. Telecommunications Science, 2009, 25(2): 28-34.
- [2] 王帅, 汪来富, 金华敏, 等. 网络安全分析中的大数据技术应用[J]. 电信科学, 2015, 31(7): 145-150.
WANG S, WANG L F, JIN H M, et al. Big data application in network security analysis [J]. Telecommunications Science, 2015, 31(7): 145-150.
- [3] 万威, 赵险峰, 黄炜, 等. 基于码表分布特征和重编码的MP3Stego隐写分析[J]. 中国科学院大学学报, 2012, 29(1): 118-124.
WAN W, ZHAO X F, HUANG W, et al. Steganalysis of MP3Stego based on Huffman table distribution and recoding[J]. Journal of the Graduate School of the Chinese Academy of Sciences, 2012, 29(1): 118-124.
- [4] 万威. MP3音频隐写分析技术研究[D]. 北京: 中国科学院大学, 2011.
WAN W. Research on MP3 audio steganalysis technology[D]. Beijing: Chinese Academy of Sciences University, 2011.
- [5] PETITCOLAS F A P. MP3Stego [EB]. 2002.
- [6] ACHMAD Z. MP3Stegz [EB]. 2007.
- [7] PLATT C. UnderMP3Cover [EB]. 2002.
- [8] FRIDRICH J, GOLJAN M, DU R. Detecting LSB steganography in color, and gray-scale images[J]. IEEE Multimedia, 2001, 8(4): 22-28.
- [9] 张涛, 平西建. 基于差分直方图实现LSB信息伪装的可靠检测[J]. 软件学报, 2004, 15(1): 151-158.
ZHAN T, PING X J. Reliable detection of spatial LSB steganography based on difference histogram[J]. Journal of Software, 2004, 15(1): 151-158.
- [10] JIN C, WANG R, YAN D, et al. Steganalysis of UnderMP3Cover [J]. Journal of Computational Information Systems, 2012, 8(24): 10459-10468.
- [11] KER A D. Improved detection of LSB steganography in grayscale images[C]// International Conference on Information Hiding, May 19-21, 2004, Santa Barbara, California, USA. Berlin: Springer, 2004: 97-115.
- [12] KER A D. Quantitative evaluation of pairs and RS steganalysis[J]. Proceedings of SPIE -The International Society for Optical Engineering, 2004(5306): 83-97.
- [13] LUO X, LIU B, LIU F. Improved RS method for detection of LSB steganography[C]//International Conference on Computational Science and Its Applications, Dec 19-21, 2008, Wuhan, China. Berlin: Springer Berlin Heidelberg, 2005: 508-516.
- [14] 丁晋晶, 蒋铃鸽, 何晨. 一种针对LSB嵌入方式水印的改进的全盲检测方案[J]. 高技术通讯, 2007, 17(7): 681-685.
DING J J, JIANG L G, HE C. An improved method of detecting LSB steganography[J]. Chinese High Technology Letters, 2007, 17(7): 681-685.
- [15] 于祺, 陈克非. RS检测算法的分析与改进[J]. 计算机工程, 2008, 34(8): 170-171.
YU Z, CHEN K F. Analysis and improvement on RS detection algorithm[J]. Computer Engineering, 2008, 34(8): 170-171.

[作者简介]



张坚(1993-), 男, 宁波大学信息科学与工程学院硕士生, 主要研究方向为多媒体通信与信息安全等。



王让定(1962-), 男, 博士, 宁波大学高等技术研究院教授、博士生导师, 主要研究方向为多媒体通信与取证、信息隐藏与隐写分析、智能抄表及传感网络技术等。

严迪群(1979-), 男, 博士, 宁波大学信息科学与工程学院副教授、硕士生导师, 主要研究方向为多媒体通信、信息安全、基于深度学习的数字语音取证等。