基于随机线性码的快速矩阵嵌入方法

高瞻瞻[1],韦大伟[1],汤光明[1],李晓利[2]
(1. 解放军信息工程大学,河南郑州 450001; 2. 中国洛阳电子装备试验中心,河南洛阳 471000)

摘 要: 矩阵嵌入将编码思想引入隐写过程,用病灶携带秘密信息,通过寻找校验矩阵的陪集首确定最小修改向量,提高隐写安全性.如何以较低的计算复杂度找到陪集首是矩阵嵌入设计的核心.针对小嵌入率下的隐写,该文讨论了将汉明码矩阵引入到随机线性码矩阵的可行性,进而提出了一种新的校验矩阵结构.在此基础上,以一定的计算复杂度限制为前提,以最大化嵌入效率为目标,给出了矩阵的最优化构造方法.实验结果表明,新方法相比已有矩阵嵌入方法在嵌入效率和嵌入速度上都有所提高,适合实时性要求高的隐写应用.

关键词: 隐写术; 矩阵嵌入; 嵌入效率; 嵌入速度
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2017)05-1139-11
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.05.017

Fast Matrix Embedding Based on Random Linear Code

GAO Zhan-zhan[1], WEI Da-wei[1], TANG Guang-ming[1], LI Xiao-li[2]
(1. *PLA Information and Technology University, Zhengzhou, Henan* 450001, *China*;
2. *Luoyang Electronic Equipment Test Center of China, Luoyang, Henan* 471000, *China*)

Abstract: Matrix embedding (ME) encodes cover objects and secret messages with an error correction code and modifies the cover data according to the coding result. The modification vector is the coset leader of error correction codes and the messages are conveyed by the syndrome. How to find the coset leader with lower computational complexity is the core of ME design. To improve the embedding efficiency and embedding speed of small payloads ME, this paper discusses the feasibility of introducing Hamming codes into the parity check matrix (PCM) of random linear codes, and proposes a new PCM structure. On this basis, an optimization scheme is proposed which can adaptively generate a PCM to accommodate to the given cover length and provide the best performance while guaranteeing the desired computational complexity. Experimental results show that this new method achieves higher embedding efficiency and faster embedding speed than previous fast ME methods, and is more suitable for real-time steganographic systems.

Key words: steganography; matrix embedding; embedding efficiency; embedding speed

1 引言

隐写是一种将秘密信息存储到图像、音频、视频等常见载体中以实现隐蔽通信的技术.隐写术研究的目标是提高秘密信息的隐蔽性,减小信息嵌入对载体的影响.目前,主要有以下两种方式:(1)优先选择载体中不宜感知的部分嵌入秘密信息;(2)隐藏秘密信息时尽可能少地修改载体元素.第一类方法涉及的是自适应隐写技术[1],第二类方法涉及的是嵌入效率问题[2~5].嵌入效率指修改一比特载体元素平均嵌入的信息量.嵌入率一定时,高的嵌入效率意味着较少的载体修改,因而有助于安全性的提高.

矩阵嵌入用线性码的陪集表示载密体序列,用病灶携带秘密信息,通过寻找检验矩阵(parity check matrix,PCM)的陪集首来减少修改量,从而给出了提高嵌入效率的有效途径.该思想最早由 Crandall[2] 提出,Westfeld 将汉明码应用于 F5 算法[3],使该方法广为人知.之后,Fridrich 研究了矩阵编码嵌入效率的理论上限[4,5],并证明使用二元随机线性码的矩阵嵌入可以达到该极限值.如今,矩阵嵌入已扩展到卷积码,如 Filler 等提出的量化格子编码(syndrome trellis codes, STCs)[6,7].

收稿日期:2015-12-21;修回日期:2016-10-03;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61272488);河南省科技攻关项目(No.122102210047);全军军事类研究生资助课题(No.2015JY125)

嵌入效率通常会随着 PCM 维数的增大而提高,但计算复杂度也会相应增加. 实际上,当 PCM 为随机矩阵时,寻找其陪集首是一个 NP 困难问题. 为解决这一问题,研究者主要从两方面对矩阵编码进行改进:其一是构造具有特殊形式的 PCM. 如基于汉明码的嵌入方法[3]以及 Fridrich 基于随机线性码设计的嵌入方法[8]. 这两种方法均具有优秀的矩阵结构,被后续研究者广泛关注和引用. 针对汉明码,Mao 等[9]调整了 PCM 各列的次序,利用查找表寻找陪集首,计算复杂度只有 O(1);Tian 等[10]改造了汉明码 PCM 的结构,克服了汉明码只能实现个别嵌入量的缺点. 针对 Fridrich 的方法,Wang 等[11]提出将 PCM 中部分随机列转化为具有特定形式的参考列,保证嵌入效率的同时有效降低了计算复杂度. 另一类改进方法是寻找次优修改向量代替陪集首. 为此,Gao 等[12]从陪集中挑选汉明重量相对较小的向量对载体进行修改,该方法降低了计算复杂度但一定程度上损害了嵌入效率. 与此相近的还有文献[13,14]提出的嵌入方法.

受文献[11]的启发,本文对随机线性码 PCM 的优化构造作了进一步研究. 针对小嵌入率(嵌入率低于 0.5)隐写,给出了可实现快速嵌入的矩阵编码方法. 实验结果表明,PCM 大小相同时,本文方法相比已有编码方式有效提高了嵌入效率,并降低了计算复杂度.

2 基于随机线性码的矩阵嵌入

2.1 矩阵嵌入

令维数为 $(n-k)\times n$ 的 $\boldsymbol{H}$ 表示 $[n,k]$ 线性码 C 的 PCM,$\boldsymbol{c}^{\mathrm{T}}=(c_1,c_2,\cdots c_n)\in GF^n(2)$ 表示载体元素集合,秘密信息为 $\boldsymbol{m}^{\mathrm{T}}=(m_1,m_2,\cdots m_m)\in GF^m(2)$. 为使嵌入引起的修改量最小,首先对 $\boldsymbol{m}$ 和 $\boldsymbol{Hc}$ 进行按位异或运算,得到病灶 $\boldsymbol{u}=\boldsymbol{m}\oplus\boldsymbol{Hc},\boldsymbol{u}\in F_2^{n-k}$. 进而得到其在 $\boldsymbol{H}$ 下的陪集:

$$C_{\boldsymbol{H}}(\boldsymbol{u})=\{\boldsymbol{x}\in GF_2^n\mid\boldsymbol{Hx}=\boldsymbol{u}\} \tag{1}$$

每个陪集 $C_{\boldsymbol{H}}(\boldsymbol{u})$ 含有 2^K 个向量,其中汉明重量最小的向量称为 $C_{\boldsymbol{H}}(\boldsymbol{u})$ 的陪集首,用 $e_L(\boldsymbol{u})$ 表示

$$e_L(\boldsymbol{u})=\arg\min_{\boldsymbol{x}\in C_{\boldsymbol{H}}(\boldsymbol{u})}\omega(\boldsymbol{x}) \tag{2}$$

最终的载密体 $\boldsymbol{s}$ 通过下式得到:

$$\boldsymbol{s}=\boldsymbol{c}\oplus e_L(\boldsymbol{u}) \tag{3}$$

接收端用 $\boldsymbol{H}$ 左乘 $\boldsymbol{s}$ 即可正确提取秘密信息:

$$\begin{aligned}\boldsymbol{Hs}&=\boldsymbol{H}(\boldsymbol{c}\oplus e_L(\boldsymbol{u}))=\boldsymbol{Hc}\oplus\boldsymbol{H}\cdot e_L(\boldsymbol{u})\\&=\boldsymbol{Hc}\oplus(\boldsymbol{m}\oplus\boldsymbol{Hc})=\boldsymbol{m}\end{aligned} \tag{4}$$

2.2 基于随机线性码的矩阵嵌入

Fridrich 等最早提出基于 $[n,k]$ 随机线性码的矩阵嵌入方法[8]. 该方法的 PCM 基本形式如下:

$$\boldsymbol{H}=(\boldsymbol{I}_{n-k},\boldsymbol{R}) \tag{5}$$

其中,$\boldsymbol{I}_{n-k}$ 表示大小为 $(n-k)\times(n-k)$ 的单位矩阵,$\boldsymbol{R}$ 表示 $(n-k)\times k$ 的随机矩阵. 病灶 $\boldsymbol{u}$ 的陪集大小为 2^k,对应 2^k 种随机列组合. 当嵌入率较大时,k 值相对较小,通过穷举法即可找到陪集首,计算复杂度为 $O(n2^k)$.

为进一步降低计算复杂度,Wang 等提出基于参考列的快速矩阵嵌入方法[11],其矩阵形式为:

$$\boldsymbol{H}=(\boldsymbol{I}_{n-k},\boldsymbol{R},\boldsymbol{D}) \tag{6}$$

其中,$\boldsymbol{R}$ 为 $(n-k)\times k_1$ 的随机矩阵,矩阵 $\boldsymbol{D}$ 形式固定,大小为 $(n-k)\times k_2,k_1+k_2=k$. $\boldsymbol{D}$ 中第 i 列

$$\boldsymbol{d}_i^{\mathrm{T}}=(\boldsymbol{0}_{t_1},\cdots,\boldsymbol{0}_{t_{i-1}},\boldsymbol{1}_{t_i},\boldsymbol{0}_{t_{i+1}},\cdots,\boldsymbol{0}_{t_{k_2}}),1\leqslant i\leqslant k_2 \tag{7}$$

其中,t_i 的取值为:

$$t_i=\begin{cases}\lfloor\dfrac{n-k}{k_2}\rfloor, & \text{if } i<k_2\\ (n-k)-(k_2-1)\lfloor\dfrac{n-k}{k_2}\rfloor, & \text{if } i=k_2\end{cases} \tag{8}$$

例如,参数 $(n,k,k_1,k_2)=(11,5,2,3)$ 时,由式(8)知 $t_1=t_2=t_3=2$,$\boldsymbol{H}$ 的具体形式为:

$$\begin{pmatrix}1&0&0&0&0&0&0&1&1&0&0\\0&1&0&0&0&0&1&0&1&0&0\\0&0&1&0&0&0&1&1&0&1&0\\0&0&0&1&0&0&0&0&0&1&0\\0&0&0&0&1&0&1&1&0&0&1\\0&0&0&0&0&1&0&1&0&0&1\end{pmatrix} \tag{9}$$

根据 $\boldsymbol{H}$ 的结构,将修改向量 $\boldsymbol{x}^{\mathrm{T}}$ 写成 $(\boldsymbol{x}_0^{\mathrm{T}},\boldsymbol{x}_1^{\mathrm{T}},\boldsymbol{x}_2^{\mathrm{T}})$ 的形式,则有

$$\boldsymbol{x}_0\oplus\boldsymbol{R}\boldsymbol{x}_1\oplus\boldsymbol{D}\boldsymbol{x}_2=\boldsymbol{u} \tag{10}$$

故,$\boldsymbol{x}_0\oplus\boldsymbol{D}\boldsymbol{x}_2=\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1$. 将 $\boldsymbol{x}_0$ 和 $\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1$ 都分成 k_2 部分,并保证各部分的长度 $|\boldsymbol{x}_{0,i}|=|(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i|=t_i$. 依据 $\boldsymbol{x}_2=(x_{2,1},\cdots,x_{2,k_2})$ 内各元素取值的不同,$\boldsymbol{x}_{0,i}$ 的取值有两种情况:

$$\boldsymbol{x}_{0,i}=\begin{cases}(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i, & \text{if } x_{2,i}=0\\ \boldsymbol{1}_{t_i}\oplus(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i, & \text{if } x_{2,i}=1\end{cases} \tag{11}$$

因此,$\boldsymbol{x}_{0,i}$ 的汉明重量为

$$\omega(\boldsymbol{x}_{0,i})=\begin{cases}\omega(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i & \text{if } x_{2,i}=0\\ t_i-\omega(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i & \text{if } x_{2,i}=1\end{cases} \tag{12}$$

最终,确定 $C_{\boldsymbol{H}}(\boldsymbol{u})$ 的陪集首转化为寻找使下式最小的修改向量:

$$\begin{aligned}\omega(\boldsymbol{x}_1)+\sum_{i=1}^{k_2}\min\{&\omega((\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i),t_i\\&-\omega(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i+1)\}\end{aligned} \tag{13}$$

3 实现快速矩阵嵌入的方法分析

调整式(6)$\boldsymbol{H}$ 内各列的位置,Wang 等构造的矩阵可以转化为如下形式:

$$\boldsymbol{H}=(\boldsymbol{I}_{n-k},\boldsymbol{R},\boldsymbol{D})=(\boldsymbol{A},\boldsymbol{R}) \tag{14}$$

其中,$\boldsymbol{A}$ 是一个 $(n-k)\times(n-k+k_2)$ 的矩阵,它包含了所有的非随机列. 如式(9)所示矩阵可写为:

$$\boldsymbol{A}=\begin{pmatrix}\boldsymbol{B}_1 & 0 & 0\\ 0 & \boldsymbol{B}_2 & 0\\ 0 & 0 & \boldsymbol{B}_3\end{pmatrix},\boldsymbol{B}_1=\boldsymbol{B}_2=\boldsymbol{B}_3=\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 1\end{pmatrix} \tag{15}$$

将修改向量分成两部分 $\boldsymbol{x}^{\mathrm{T}}=(\boldsymbol{x}_0^{\mathrm{T}},\boldsymbol{x}_1^{\mathrm{T}})$,则:

$$((\boldsymbol{B}_1\boldsymbol{x}_{0,1})^{\mathrm{T}},(\boldsymbol{B}_2\boldsymbol{x}_{0,2})^{\mathrm{T}},\cdots,(\boldsymbol{B}_{k_2}\boldsymbol{x}_{0,k_2})^{\mathrm{T}})^{\mathrm{T}}\oplus\boldsymbol{R}\boldsymbol{x}_1=\boldsymbol{u} \tag{16}$$

由于局部矩阵 $\boldsymbol{A}$ 的特殊结构,有下式成立:

$$\boldsymbol{B}_i\boldsymbol{x}_{0,i}=(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)_i \tag{17}$$

$\boldsymbol{B}_i=(\boldsymbol{I}_{t_i},\boldsymbol{1}_{t_i})$,形式简单易于计算,因此利用式(12)可以迅速找到满足上式且汉明重量最小的局部修改向量 $\boldsymbol{x}_{0,i}^{opt}$. 遍历 $\boldsymbol{x}_1$ 的 2^{k_1} 种取值,总汉明重量 $\sum_{i=1}^{k_2}\omega(\boldsymbol{x}_{0,i}^{opt})+\omega(\boldsymbol{x}_1)$ 最小时的修改向量即为陪集首,其形式为:$((\boldsymbol{x}_{0,1}^{opt})^{\mathrm{T}},\cdots,(\boldsymbol{x}_{0,k_2}^{opt})^{\mathrm{T}},\boldsymbol{x}_1^{\mathrm{T}})^{\mathrm{T}}$. 相比 Fridrich 的方法[7],计算复杂度由 $O(n2^k)$ 降为 $O(n2^{k_1})$.

由上述分析可见,若大小为 $(n-k)\times n$ 的 PCM 中随机列个数小于 k,求 $C_{\boldsymbol{H}}(\boldsymbol{u})$ 的陪集首需分两步进行:首先得到 $C_{\boldsymbol{A}}(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)$ 在不同 $\boldsymbol{x}_1$ 下的陪集首;之后从中选出 $\omega(C_{\boldsymbol{A}}(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1))+\omega(\boldsymbol{x}_1)$ 最小的一组.

矩阵嵌入的计算复杂度与随机列个数指数相关,为降低编码开销,随机列个数通常小于 k. 这种情况下,矩阵编码的性能主要受局部矩阵 $\boldsymbol{A}$ 影响. 因此,需要合理设计其矩阵结构.

4　快速矩阵嵌入方法

4.1　校验矩阵的结构

对于 Wang 等的快速矩阵嵌入方法,当 k_2 较小时,新增的参考列可有效提高矩阵 $\boldsymbol{A}$ 的嵌入效率,进而带来整体性能的改善. 但随着 k_2 的增大,参考列对 $\boldsymbol{A}$ 的影响逐渐减弱,$\lfloor(n-k)/k_2\rfloor\geqslant 1$ 时(嵌入率低于 0.5)已不会提高 $\boldsymbol{A}$ 的嵌入效率,这说明该方法不适用于小嵌入率隐写. 但是,我们发现:当 $\lfloor(n-k)/k_2\rfloor=2$ 时,$\boldsymbol{A}$ 的子矩阵 $\boldsymbol{B}$ 其实是[3,1]汉明码的形式(如式(15)所示). 汉明码是一种优秀的隐写编码,在小嵌入率下的嵌入效率较高,且经文献[9]的改进后计算复杂度只有 $O(1)$,将其引入矩阵 $\boldsymbol{A}$ 有助于快速求出 $C_{\boldsymbol{A}}(\boldsymbol{u}\oplus\boldsymbol{R}\boldsymbol{x}_1)$ 的陪集首. 因此,可以借助汉明码来进一步改善 Wang 等的方法的适用范围. 本文构造的 PCM 基本形式如下所示:

$$\boldsymbol{H}=(\boldsymbol{A},\boldsymbol{R}),\ \boldsymbol{A}=\begin{pmatrix}\boldsymbol{B}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{B}_2 & & \boldsymbol{0}\\ \vdots & & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{B}_p\end{pmatrix} \tag{18}$$

其中,$\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_p$ 等均为汉明码对应的 PCM,它们可以相同也可以不同.

4.2　校验矩阵的最优化

给定秘密信息 $\boldsymbol{m}$ 和载体 $\boldsymbol{c}$,$\boldsymbol{H}$ 的大小也就随之确定. 给定计算复杂度限制,则 $\boldsymbol{H}$ 中随机列的个数也基本确定. 因此本部分就如何确定 $\boldsymbol{A}$ 的具体结构,从嵌入速度和嵌入效率两方面展开讨论.

用 p 表示 $\boldsymbol{A}$ 中子矩阵个数,子矩阵 $\boldsymbol{B}_i$ 的行高和列宽分别记为 $r_i,w_i,i\in\{1,2,\cdots,p\}$. 汉明码中 w 与 r 有固定的函数关系,记为 f:

$$w_i=f(r_i)=2^{r_i}-1,i\in\{1,2,\cdots,p\} \tag{19}$$

因此,$\boldsymbol{A}$ 的结构可用行高的集合 $\boldsymbol{r}=\{r_1,r_1\cdots,r_p\}$ 表示. 根据式(19),可得如下定理:

定理 1　若局部矩阵 $\boldsymbol{A}$ 包含 p 个子矩阵,且行高 $n-k$ 固定,则其列宽的取值范围为:

$$p(2^{\frac{n-k}{p}}-1)=pf(\frac{n-k}{p})\leqslant\sum_{i=1}^{p}w_i\leqslant p+2^{n-k-(p-1)}-2 \tag{20}$$

证明　$n-k$ 固定,故有

$$\sum_{i=1}^{p}r_i=n-k,r_i\in N^*$$

由均值不等式得:

$$\begin{aligned}\sum_{i=1}^{p}w_i&=f(r_1)+f(r_2)+\cdots+f(r_p)\\&=(2^{r_1}-1)+(2^{r_2}-1)+\cdots+(2^{r_p}-1)\\&=2^{r_1}+2^{r_2}+\cdots+2^{r_p}-p\\&\geqslant p\sqrt[p]{2^{r_1+r_2+\cdots+r_p}}-p\\&=p(2^{\frac{n-k}{p}}-1)\end{aligned}$$

当且仅当 $r_1=\cdots=r_p=\dfrac{(n-k)}{p}$ 时,等号成立.

另一方面,$r\geqslant 1$ 时,有 $f'(r)=2^r\ln 2>1$,且 $f''(r)=2^r(\ln 2)^2>0$,因此 $f(r)$ 随着 r 的增大而增大,且增幅逐步加快. 故

$$\begin{aligned}\sum_{i=1}^{p}w_i&=f(r_1)+f(r_2)+\cdots+f(r_p)\\&\leqslant f(1)+f(r_2)+\cdots+f(r_p+(r_1-1))\leqslant\cdots\\&\leqslant f(1)+f(1)+\cdots+f(r_p+(r_1-1)+\cdots(r_{p-1}-1))\\&=f(1)+f(1)+\cdots+f(n-k-(p-1))\\&=p+2^{n-k-(p-1)}-2\end{aligned}$$

证毕.

通过以上证明过程可以发现:$\boldsymbol{A}$ 的列宽与子矩阵间的差异性相关,差异性越大列宽越大. 当各子矩阵的大小同为 $(n-k)/p$ 时 $\boldsymbol{A}$ 的列宽最小,当 $r=\{1,1,\cdots,1,n-k-(p-1)\}$ 时列宽最大.

定理 2　若局部矩阵 $\boldsymbol{A}$ 的行高 $n-k$ 固定,则子矩阵个数越少(p 越小)越有利于增大 $\boldsymbol{A}$ 的列宽.

证明　$\boldsymbol{A}$ 包含 p 个子矩阵时,记 $\boldsymbol{r}=\{r_1,\cdots r_p\}$,$\boldsymbol{A}$ 包含 p' 个子矩阵时,记 $\boldsymbol{r}'=\{r_1',\cdots,r_{p'}'\}$. 若 $p'=p+1$,根

据定理 1 有下式成立：

$$\begin{aligned}\max\sum_{i=1}^{p} w_i &= p+2^{n-k-(p-1)}-2\\ &= p+1+2^{n-k-p}-2+2^{n-k-p}-1\\ &\geqslant p+1+2^{n-k-p}-2\\ &= \max\sum_{i=1}^{p'} w_i'\end{aligned}$$

$\min\sum_{i=1}^{p} w_i = p(2^{(n-k)/p}-1)$，以 p 为自变量对函数 $\min\sum_{i=1}^{p} w_i$ 求一阶导数：

$$\begin{aligned}\frac{\mathrm{d}(\min\sum_{i=1}^{p} w_i)}{\mathrm{d}p} &= 2^{\frac{n-k}{p}}-1-p\cdot 2^{\frac{n-k}{p}}\ln 2\cdot\frac{n-k}{p^2}\\ &= 2^{\frac{n-k}{p}}\left(1-\ln 2\cdot\frac{n-k}{p}\right)-1\end{aligned}$$

易知当 $\frac{(n-k)}{p}\geqslant 1$ 时，$2^{\frac{n-k}{p}}\left(1-\ln 2\cdot\frac{n-k}{p}\right)<1$，因此 $\frac{\mathrm{d}(\min\sum_{i=1}^{p} w_i)}{\mathrm{d}p}<0$，即

$$\min\sum_{i=1}^{p} w_i > \min\sum_{i=1}^{p'} w_i'$$

证毕.

汉明码可以在 2^r-1 比特的载体序列中嵌入 r 比特的秘密信息，且最多引起 1 比特的改变. 载体改变 1 比特的概率为 $(2^r-1)/2^r$，不改变的概率为 $1/2^r$. 因此，一个 $[2^r-1,2^r-1-r]$ 汉明码的嵌入效率为 $r\cdot 2^r/(2^r-1)$. 据此可推出如下定理：

定理 3 若局部矩阵 $\boldsymbol{A}$ 包含 p 个子矩阵，且行高 $n-k$ 固定，则其嵌入效率与子矩阵间的差异性相关，差异性越大嵌入效率越高，嵌入效率的取值范围为：

$$\frac{n-k}{p-p\dfrac{1}{2^{n-k/p}}}\leqslant e\leqslant\frac{n-k}{\dfrac{p+1}{2}-\dfrac{1}{2^{n-k-(p-1)}}} \tag{21}$$

证明 局部矩阵 $\boldsymbol{A}$ 包含 p 个子矩阵，故嵌入信息后载体的平均修改量

$$E(\omega_{\boldsymbol{A}}) = \sum_{i=1}^{p}\frac{2^{r_i}-1}{2^{r_i}} = p-\sum_{i=1}^{p}\frac{1}{2^{r_i}}$$

记 $g(r)=-2^{-r}$. 由均值不等式得：

$$\begin{aligned}\sum_{i=1}^{p} g(r_i) &= -2^{-r_1}-2^{-r_2}\cdots-2^{-r_p}\\ &\leqslant -p\sqrt[p]{2^{-r_1-r_2-\cdots-r_p}}\\ &= -p\cdot 2^{-\frac{n-k}{p}}\end{aligned}$$

故 $E(\omega_{\boldsymbol{A}})\leqslant p-p/2^{(n-k)/p}$.

当且仅当 $r_1=\cdots=r_p=(n-k)/p$ 时，等号成立.

另一方面，$r\geqslant 1$ 时，有 $g'(r)=2^{-r}\ln 2>0$，且 $g''(r)=-2^{-r}(\ln 2)^2<0$，因此 $g(r)$ 随着 r 的增大而增大，但增幅逐步放缓. 故

$$\begin{aligned}\sum_{i=1}^{p} g(r_i) &= g(r_1)+g(r_2)+\cdots+g(r_p)\\ &\geqslant g(1)+g(r_2)+\cdots+g(r_p+(r_1-1))\geqslant\cdots\\ &\geqslant g(1)+g(1)+\cdots+g(r_p+(r_1-1)\\ &\quad+\cdots(r_{p-1}-1))\\ &= g(1)+g(1)+\cdots+g(n-k-(p-1))\end{aligned}$$

因此，

$$E(\omega_{\boldsymbol{A}})\geqslant p-\left(\frac{p-1}{2}+\frac{1}{2^{n-k-(p-1)}}\right)=\frac{p+1}{2}-\frac{1}{2^{n-k-(p-1)}}$$

秘密信息长度 $n-k$ 除以平均失真 $E(\omega_{\boldsymbol{A}})$ 即得式(21)所示嵌入效率.

证毕.

定理 4 若局部矩阵 $\boldsymbol{A}$ 的行高 $n-k$ 固定，则子矩阵个数越少(即 p 越小)越有利于提高隐写的嵌入效率.

以定理 3 为基础进行证明，具体过程参考定理 2 的证明，此处不再赘述.

定理 2 和定理 4 说明：为提高嵌入速度和嵌入效率，矩阵的优化构造应首先以最小化子矩阵个数为目标. 定理 1 和定理 3 说明：子矩阵个数一定时，增大子矩阵间的差异性，不仅能减小随机列的个数，降低计算量，也有助于提高嵌入效率. 例如，要将 5 比特秘密信息嵌入到 20 比特的载体序列中. 可以令 $r_1=2, r_2=3$，采用[3,1]码和[7,4]码的组合方式；也可以令 $r_1=1, r_2=4$，采用一阶单位阵和[15,11]码的组合方式. 但前者的 PCM 中会额外产生 10 个随机列，计算复杂度过大，而后者只有 4 个的随机列. 并且第一种方式下 $\boldsymbol{A}$ 的嵌入效率为 28/9，低于后者的 80/23.

综上，局部矩阵 $\boldsymbol{A}$ 的最优化构造分两步进行：

首先，根据定理 1，寻找取值最小且满足不等式 $p(2^{(n-k)/p}-1)\leqslant n\leqslant p+2^{n-k-(p-1)}-2$ 的 p，即解决以下优化问题：

$$\begin{cases}\text{minimize} & p\\ \text{subject to} & l\lfloor\dfrac{n-k}{p}\rfloor+(p-l)\lfloor\dfrac{n-k}{p}\rfloor=n-k\\ & p\in N^*,\quad l\in\{1,2,\cdots,p\}\\ & l(2^{\lfloor\frac{n-k}{p}\rfloor}-1)+(p-l)(2^{\lfloor\frac{n-k}{p}\rfloor}-1)\leqslant n\\ & p+2^{n-k-(p-1)}-2\geqslant n\end{cases} \tag{22}$$

具体步骤如下：

Step 1 初始化 $p=1$；

Step 2 依下式确定参数 l：

$$l\lfloor\frac{n-k}{p}\rfloor+(p-l)\lfloor\frac{n-k}{p}\rfloor=n-k,\quad l\in\{1,2,\cdots,p\}$$

Step 3 若 $l(2^{\lfloor (n-k)/p \rfloor}-1)+(p-l)(2^{\lfloor (n-k)/p \rfloor}-1) \leqslant n$,进入下一步. 否则,$p=p+1$,返回 Step 2;

Step 4 输出 p.

p 值确定后即可确定各子矩阵的大小,这个过程应以最大化子矩阵间的差异性为优化目标:

$$\begin{cases} \text{maximize} & \sum_{i=1}^{p} w_i = 2^{r_1}+2^{r_2}+\cdots+2^{r_p}-p \\ \text{subject to} & \sum_{i=1}^{p} r_i = n-k \\ & \sum_{i=1}^{p} w_i \leqslant n \\ & r_i \in N^*, i \in \{1,2,\cdots,p\} \end{cases} \tag{23}$$

具体步骤如下:

Step 1 初始化,令 $r_p=n-k$,$r_1=\cdots=r_{p-1}=1$,$j=p$,$p_e=1$;

Step 2 若 $j \leqslant p-p_e$,$p_e=p_e+1$,$j=p-1$;

Step 3 若 $\sum_{i=1}^{p}(2^{r_i}-1) \leqslant n$,转到 Step 5. 否则进入下一步;

Step 4 $r_p=r_p-1$. 若 $r_j < \lceil (n-k-(p-p_e))/p_e \rceil$,$r_j=r_j+1$ 并返回 Step 3. 否则,$j=j-1$,$r_j=r_j+1$ 并返回 Step 2;

Step 5 令 $k_2=\sum_{i=1}^{p}(2^{r_i}-1)-(n-k)$,$k_1=k-k_2$,输出参数 $r_1,r_2,\cdots,r_p,k_1,k_2$.

4.3　陪集首的快速求解

根据第 3 节的分析,分两步寻找陪集首. 为加快第一步的求解,迅速得到 $C_A(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1)$ 在不同 $\boldsymbol{x}_1$ 下的陪集首,本文参考文献[9]调整子矩阵 $\boldsymbol{B}_i$ 内各列的位置,使其按递增(或递减)的大小关系排列,如下所示:

$$\boldsymbol{B}_1=\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{24}$$

此时,$(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1)_i$ 自身指示了载体元素要修改的位置. 例如$(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1)_1^{\mathrm{T}}=(1,0,0)$,其代表的十进制数为 4,则最优局部修改向量 $\boldsymbol{x}_{1,1}^{\mathrm{T}}=(0,0,0,1,0,0,0)$,计算复杂度只有 $O(1)$. $\boldsymbol{A}$ 内包含 p 个子矩阵,故求 $C_A(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1)$ 的陪集首的计算复杂度为 $O(p)$.

定理 5　由汉明码的嵌入特点可知:如果矩阵 $\boldsymbol{A}$ 包含 p 个子矩阵,那么无论 $\boldsymbol{x}_1$ 取值如何,总有 $\omega(C_A(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1)) \leqslant p$ 成立. 因此,最优的修改向量 $\boldsymbol{x}$ 满足下式:

$$\omega(C_A(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1))+\omega(\boldsymbol{x}_1) \leqslant p \tag{25}$$

根据定理 5,对于陪集首必然有 $\omega(\boldsymbol{x}_1) \leqslant p$ 成立. 这一结论允许我们在第二步不必遍历所有的随机列组合. 具体来说,算法只需在 $\omega(\boldsymbol{x}_1)<p$ 时求出其对应的修改向量,计算这些向量的汉明重量 $\omega(C_A(\boldsymbol{u} \oplus \boldsymbol{R}\boldsymbol{x}_1))+\omega(\boldsymbol{x}_1)$,从中选出最小的一个即得陪集首. 因此,算法遍历的组合数为:

$$\mu_{k_1,p}=\begin{cases} \sum_{i=0}^{p-1}\binom{k_1}{i}, & \text{if } p-1 \leqslant k_1 \\ \sum_{i=0}^{k_1}\binom{k_1}{i}, & \text{if } p-1 > k_1 \end{cases} \tag{26}$$

总的计算复杂度为 $O(p\mu_{k_1,p})$.

4.4　秘密信息的提取

为保证接收方正确提取秘密信息,收发双方应在一次通信中共享相同的 PCM. 最简单的方式是发送方利用 LSB 等隐蔽通信方法直接将 $\boldsymbol{H}$ 传递给接收方. 除此之外,如果双方具有同步的种子密钥,发送方只需告知接收方秘密信息的长度,接收方执行相同的步骤进行矩阵的优化构造,从而得到相同的 PCM. 显然,后一种方式的隐蔽性更好. 在此基础上,接收方即可依式(4)从载密体中提取秘密信息.

5　实验结果与分析

如果某矩阵编码的计算复杂度较小,那么它可以使用较大规模的 PCM,这有助于提高嵌入效率. 因此,部分文献在比较编码嵌入效率时不考虑 PCM 的规模,仅保证相同的计算复杂度和嵌入率. 但是,实际应用中,载体可能存在分组,嵌入过程在各个分组上进行,其元素个数确定且有限,如 VoIP 隐写[10,15]. 因此,在 PCM 规模相同的情况下比较编码性能更具实际意义. 本文在接下来的实验中令各编码的 PCM 大小相同.

实验 1:随机产生二进制序列作为载体和秘密信息,检验所提方法的嵌入效率和嵌入速度. 目前的快速矩阵嵌入仍以面向大嵌入率的研究成果居多,小嵌入率下,诸如文献[8]等的方法都会由于计算复杂度过大而无法使用. 因此,本实验选择汉明码、Tian 等的编码[10]、Wang 等的编码[11]三种方法进行对比.

取载体长度 $n=60$,依 4.2 节所述步骤确定各嵌入量下的最优 PCM,如表 1 所示. 利用最优 PCM 进行信息嵌入和提取,重复 5000 次. 将接收端得到的秘密信息与初始秘密信息对比,结果表明正确率 100%. 为保证不同方法间计算复杂度的差异不致过大,实验中限定 Wang 等方法的计算复杂度不超过 $O(n2^6)$. 图 1 显示的是各编码方法在不同嵌入率下的平均嵌入效率. 结果表明:(1)相比汉明码,本文方法支持更多的信息量嵌入,通用性更好;(2)如前文所述,Wang 等的编码方法不适用于小嵌入率隐写;(3)虽然 Tian 等和 Wang 等的方法也能实现所有嵌入量下的隐写,但整体而言,其嵌入效率低于本文方法.

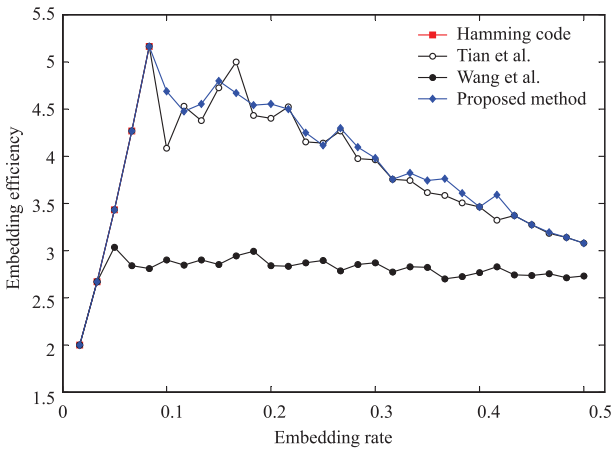


图1 本文方法与相关编码方法的嵌入效率（n=60）

实验中，汉明码、Tian 等编码的计算复杂度近似为 $O(n)$，Wang 等方法的计算复杂度为 $O(n2^6)$，本文方法的计算复杂度为 $O(p\mu_{k_1,p})$. $n=60$ 时，实际的计算复杂度如表 2 所示. 从该表可以看出，除个别情况略高于前两者外，本文方法在绝大多数嵌入量下的计算复杂度在四种方法中最小.

重复上述处理过程，得到不同载体长度下本文方法的嵌入效率，如图 2 所示. 表 3 则统计了不同嵌入率下本文方法计算复杂度的变化. 由图 2 和表 3 可知：(1)秘密信息长度相同，嵌入效率随着载体长度的增加而提高；(2)相同嵌入率下，本文方法的计算复杂度随载体长度线性增长.

表 1 各嵌入量下的最优矩阵($n=60$)

$n-k$	e	$\boldsymbol{r}$	p	k_1	$n-k$	e	$\boldsymbol{r}$	p	k_1
1	2.0150	{1}	1	59	16	4.2972	{1,3,4,4,4}	5	7
2	2.6667	{2}	1	57	17	4.0964	{2,3,4,4,4}	5	5
3	3.4416	{3}	1	53	18	3.9787	{3,3,4,4,4}	5	1
4	4.2827	{4}	1	35	19	3.7531	{1,3,3,4,4,4}	6	0
5	5.1663	{5}	1	29	20	3.8226	{3,3,3,3,4,4}	6	2
6	4.6875	{1,5}	2	28	21	3.7433	{2,2,3,3,3,4,4}	7	3
7	4.4700	{2,5}	2	26	22	3.7581	{3,3,3,3,3,3,4}	7	3
8	4.5508	{3,5}	2	22	23	3.6095	{1,3,3,3,3,3,3,4}	8	2
9	4.7974	{4,5}	2	14	24	3.4595	{2,3,3,3,3,3,3,4}	8	0
10	4.6653	{1,4,5}	3	13	25	3.5889	{1,3,3,3,3,3,3,3,3}	9	3
11	4.5371	{2,4,5}	3	11	26	3.3671	{2,3,3,3,3,3,3,3,3}	9	1
12	4.5512	{3,4,5}	3	7	27	3.2727	{1,2,3,3,3,3,3,3,3,3}	10	0
13	4.4989	{1,3,4,5}	4	6	28	3.1938	{1,2,2,2,3,3,3,3,3,3,3}	11	1
14	4.2491	{2,3,4,5}	4	4	29	3.1351	{1,1,2,2,2,3,3,3,3,3,3,3}	12	0
15	4.1126	{3,3,4,5}	4	0	30	3.0769	{2,2,2,2,2,2,3,3,3,3,3,3}	12	0

表 2 各嵌入量下的计算复杂度($n=60$)

$n-k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	1	1	1	1	1	$2\sum_{i=0}^{1}\binom{28}{i}$	$2\sum_{i=0}^{1}\binom{26}{i}$	$2\sum_{i=0}^{1}\binom{22}{i}$	$2\sum_{i=0}^{1}\binom{14}{i}$	$3\sum_{i=0}^{2}\binom{13}{i}$	$3\sum_{i=0}^{2}\binom{11}{i}$	$3\sum_{i=0}^{2}\binom{7}{i}$	$4\sum_{i=0}^{3}\binom{6}{i}$	$4\sum_{i=0}^{3}\binom{4}{i}$	4
$n-k$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
C	$5\sum_{i=0}^{4}\binom{7}{i}$	$5\sum_{i=0}^{4}\binom{5}{i}$	$5\cdot 2$	6	$6\cdot 2^2$	$7\cdot 2^3$	$7\cdot 2^3$	$8\cdot 2^2$	8	$9\cdot 2^3$	$9\cdot 2$	10	$11\cdot 2$	12	12

表 3　本文方法的计算复杂度

嵌入率	0.1	0.2	0.3	0.4	0.5
$n = 10$	1	1	1	$2\sum_{i=0}^{1}\binom{2}{i}$	2
$n = 20$	1	1	$2\sum_{i=0}^{1}\binom{2}{i}$	$3\sum_{i=0}^{2}\binom{3}{i}$	4
$n = 40$	1	$2\sum_{i=0}^{1}\binom{2}{i}$	$4 \cdot 2^2$	6	8
$n = 60$	$2\sum_{i=0}^{1}\binom{28}{i}$	$3\sum_{i=0}^{2}\binom{7}{i}$	$5 \cdot 2$	8	12
$n = 80$	$2\sum_{i=0}^{1}\binom{14}{i}$	4	$7 \cdot 2^3$	$11 \cdot 2$	16
$n = 100$	$2\sum_{i=0}^{1}\binom{22}{i}$	$5\sum_{i=0}^{4}\binom{7}{i}$	$9 \cdot 2^3$	$13 \cdot 2$	20

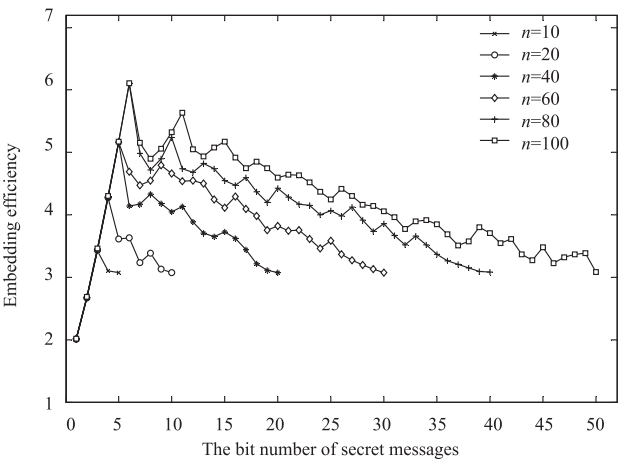


图2 本文方法的嵌入效率

实验 2:将所提编码应用于具体的隐写算法. 针对 G.723.1 6.3kbits/s 语音帧,文献[16]采用语音质量感觉评估(PESQ)[19]方法对帧中各比特的抗噪性进行测试,选取了 18 比特的最低有效位. 在此基础上,结合矩阵嵌入和 LSB 替换提出了 CLFW 算法. 本实验将该算法与所提编码相结合,实现秘密信息的隐蔽传输.

从 CMU ARCTIC 语音数据库[17]选取 1000 段语音用于实验测试,男、女声各占一半. 实验前对所有语音进行 8KHz 采样、16bit 线性 PCM 量化,转化为 G.723.1 编码器要求的输入格式. 不失一般性,秘密信息仍采用随机产生的二进制序列.

语音编/解码具有不可避免的算法时延. 语音 IP 包中的语音帧越多,编/解码器的处理时延就越大. 一般情况下,一个 IP 包至多装载 4 个 G.723.1 语音帧[18],缓存过多的语音帧会严重影响语音的传输质量. 这也就意味着:对于一次隐写操作,载体元素个数小于 $18 \times 4 = 72$. 因此,实验以载体长度 $n = 40$ 和 $n = 60$ 为例进行信息的嵌入和提取.

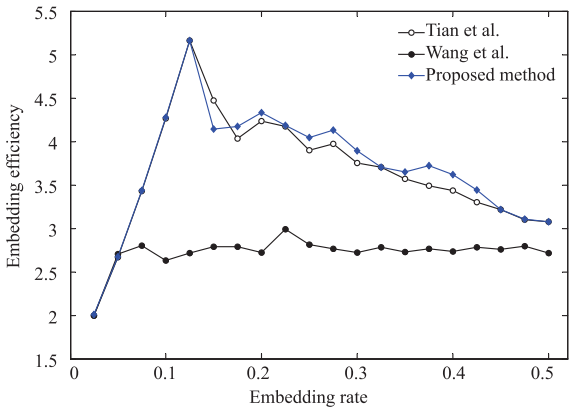
首先测试所提编码在实际应用中的嵌入效率. 分女声和男声,分别统计信息嵌入引起的元素修改个数,计算相应的嵌入效率,得到图 3 和图 4. 鉴于汉明码能实现的嵌入量有限,本实验不再与其进行对比. 同实验 1,为保证计算复杂度的差异不致过大,实验中对 Wang 等的方法的计算复杂度进行了限制:$n = 40$ 时不超过 $O(n2^5)$,$n = 60$ 时不超过 $O(n2^6)$. 由图 3、4 可知:(1)实际的嵌入效率与期望的嵌入效率相同,所提方法切实可行;(2)所提方法在三种快速矩阵嵌入方法中嵌入效率最高;(3)女声和男声的差异对嵌入效率没有影响.

PESQ[19]由 ITU(国际电信联盟)提出,是客观评价语音质量的典型方法. 选择 PESQ 衡量载密语音和原始语音间的差异,进一步检验编码的有效性.

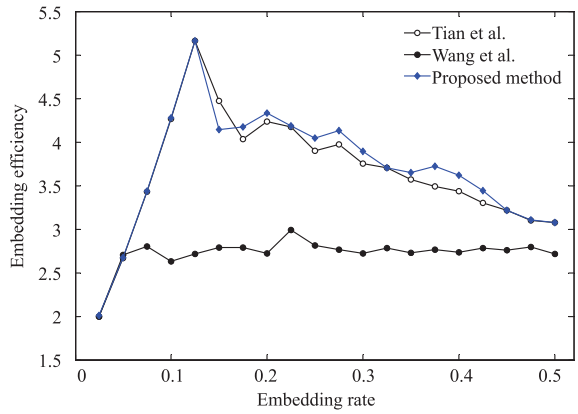
分女声、男声计算 1000 段原始语音的 PESQ 的平均值,并与不同嵌入率下载密语音的 PESQ 值进行对比,所得结果如表 4 所示. PESQ 值的取值在 -0.5(最坏)到 4.5(最好)之间,3.5 以上表示语音质量良好. 从表 4 可以看出,应用所提编码的载密语音的 PESQ 值高于其它方法,与原始语音的差值很小,说明该方法可有效保证听觉质量.

测试所提编码在嵌入速度上的性能. 为此,选择 300 段不同长度的语音,首先统计正常编/解码所用时间;然后利用不同方法实施隐写,重新测试编/解码时间;计算隐写带来的延迟. 实验中,4 个语音帧为一个语音帧组,嵌入和提取操作均在语音帧组上进行,故实际统计的是语音帧组的编/解码延迟. 其中,编码延迟的实验结果如表 5 和表 6 所示. 对于矩阵嵌入所得的载密体,信息提取只与载密体长度有关. 实验结果也表明三种方法的解码

延迟相同,$n=40$ 时三种方法的延迟同为 0.035ms,$n=60$ 时同为 0.052ms. 综合以上结果可知:(1)以秒为单位,应用本文编码的嵌入和提取延迟的数量级至多为 10^{-4},可以满足实际需要;(2)相比另外两种快速矩阵嵌入方法,本文方法具有与之相同的提取速度和更快的嵌入速度,实时性更好. 实验所用 PC 机的主要参数为:3.4GHz Intel Core i7 CPU,8GB RAM. 实验程序由 C 语言编写,在 Microsoft Visual Studio 2008 上运行.

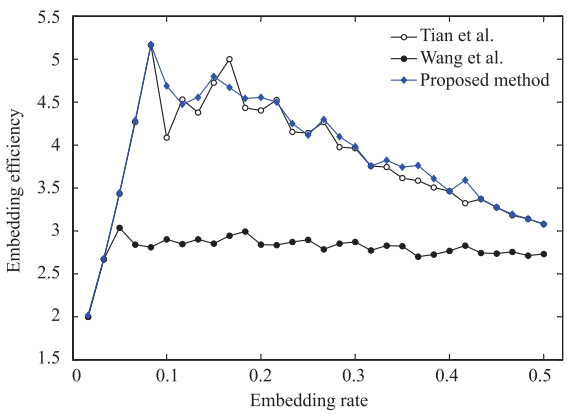


(a) 女声

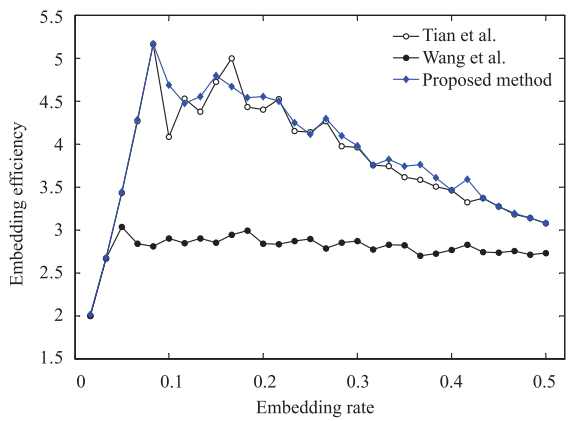


(b) 男声

图3 本文方法的实际嵌入效率(n=40)



(a) 女声



(b) 男声

图4 本文方法的实际嵌入效率(n=60)

表 4 PESQ 值对比

	女声($n=40$)			男声($n=40$)			女声($n=60$)			男声($n=60$)		
嵌入率	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5	0.1	0.3	0.5
原始语音	3.769	3.769	3.769	3.776	3.776	3.776	3.769	3.769	3.769	3.776	3.776	3.776
Tian et al.	3.738	3.567	3.426	3.745	3.574	3.432	3.687	3.531	3.346	3.694	3.538	3.352
Wang et al.	3.684	3.534	3.399	3.691	3.540	3.404	3.617	3.425	3.267	3.623	3.432	3.275
Proposed method	3.738	3.575	3.426	3.745	3.581	3.433	3.715	3.532	3.346	3.722	3.540	3.352

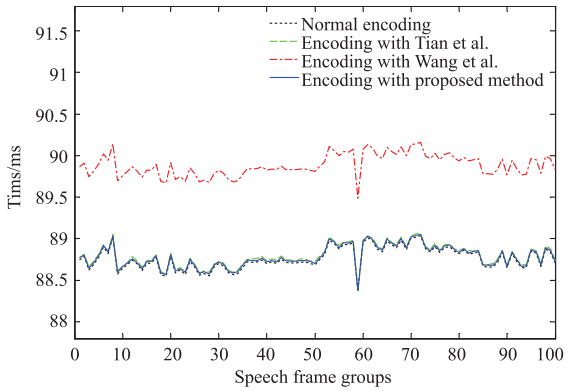
表 5 编码延迟时间对比($n=40$)

$n-k$	1	2	3	4	5	6	7	8	9	10
Tian et al.(ms)	0.001	0.002	0.004	0.008	0.015	0.027	0.017	0.019	0.024	0.031
Wang et al.(ms)	1.115	1.116	1.115	1.115	1.116	1.117	1.116	1.116	1.117	1.116
Proposed method(ms)	0.001	0.001	0.001	0.001	0.001	0.016	0.013	0.005	0.005	0.100
$n-k$	11	12	13	14	15	16	17	18	19	20
Tian et al.(ms)	0.020	0.021	0.025	0.032	0.027	0.032	0.034	0.026	0.022	0.019
Wang et al.(ms)	1.116	1.117	1.116	1.116	1.116	1.115	1.116	1.116	1.117	1.117
Proposed method(ms)	0.019	0.014	0.004	0.036	0.009	0.005	0.022	0.013	0.029	0.007

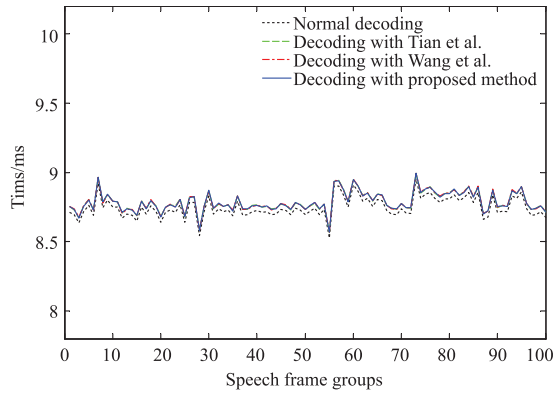
表 6　编码延迟时间对比($n=60$)

$n-k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tian et al.(ms)	0.001	0.002	0.004	0.008	0.015	0.041	0.046	0.035	0.032	0.034	0.048	0.052	0.059	0.067	0.056
Wang et al.(ms)	3.454	3.456	3.455	3.456	3.456	3.456	3.456	3.456	3.456	3.456	3.455	3.456	3.456	3.456	3.458
Proposed method(ms)	0.001	0.001	0.001	0.001	0.001	0.048	0.041	0.032	0.027	0.228	0.134	0.058	0.151	0.054	0.004
$n-k$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Tian et al.(ms)	0.060	0.051	0.054	0.058	0.042	0.055	0.052	0.049	0.043	0.061	0.057	0.043	0.045	0.031	0.028
Wang et al.(ms)	3.456	3.455	3.455	3.456	3.456	3.456	3.456	3.456	3.456	3.457	3.456	3.456	3.456	3.456	3.456
Proposed method(ms)	0.415	0.135	0.009	0.005	0.022	0.049	0.050	0.029	0.007	0.061	0.016	0.009	0.020	0.011	0.011

为更直观地进行对比,图 5 和图 6 记录了嵌入率为 0.3 时 100 个语音帧组具体的编/解码时间. 从图 5、图 6 可以看出:应用本文方法的语音编码更接近原始编码时间;本文方法下嵌入和提取操作几乎不引入额外的延迟,实时性良好.

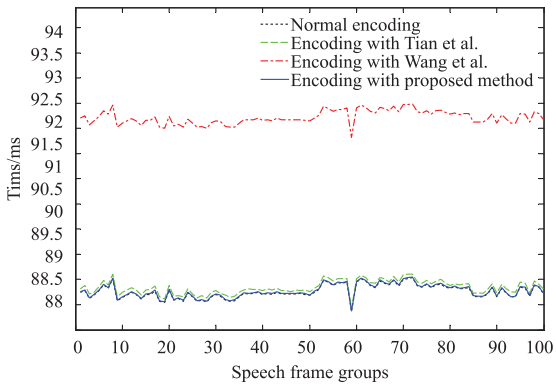


(*a*) 编码时间

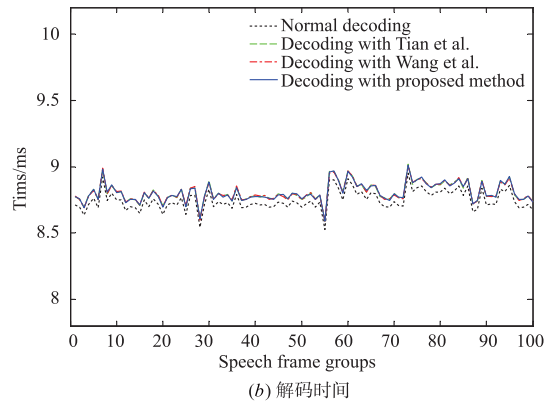


(*b*) 解码时间

图5 语音帧组的编/解码时间(n=40)



(*a*) 编码时间



(*b*) 解码时间

图6 语音帧组的编/解码时间(n=60)

实验 3:实验 1 和实验 2 说明本文方法作为一种快速矩阵嵌入方法具有较高的嵌入速度和嵌入效率. 若放宽对计算复杂度的限制,该方法的嵌入效率能得到进一步提高. PCM 中含有一定的随机列有助于提高嵌入效率,因此可以在矩阵优化前预留一定数量的随机列. 仍以 $n=60$ 为例,表 7 显示的是预留随机列后各嵌入量下本文编码的最优 PCM(限定计算复杂度不超过 $O(2^{10})$).

STCs 码[6]是最著名的卷积码嵌入方法,其计算复杂度与载体长度线性相关,在嵌入效率和嵌入速度上均有较高的性能. 图 7 记录了预留随机列时本文编码和 STCs 码的嵌入效率变化,表 8 则记录了因嵌入引起

的语音编码延迟时间. 为保证相近的计算复杂度,实验中 STCs 码的子矩阵高取为 10. 由实验结果可以看出,本文编码的嵌入效率和嵌入速度均优于 STCs 码.

表 7 各嵌入量下的最优矩阵(预留随机列)($n=60$)

$n-k$	e	$\boldsymbol{r}$	p	k_1	$n-k$	e	$\boldsymbol{r}$	p	k_1
1	2.0000	{1}	1	59	16	4.3162	{2,3,3,4,4}	5	13
2	2.6667	{2}	1	57	17	4.2252	{1,2,3,3,4,4}	6	12
3	3.4286	{3}	1	53	18	4.1247	{2,2,3,3,4,4}	6	10
4	4.2667	{4}	1	35	19	4.1408	{3,3,3,3,3,4}	6	10
5	5.1613	{5}	1	29	20	4.0638	{1,3,3,3,3,3,4}	7	9
6	4.6875	{1,5}	2	28	21	3.9626	{1,2,2,3,3,3,3,4}	8	9
7	4.4700	{2,5}	2	26	22	4.0040	{1,3,3,3,3,3,3,3}	8	10
8	4.5508	{3,5}	2	22	23	3.9159	{1,1,3,3,3,3,3,3,3}	9	9
9	4.7974	{4,5}	2	14	24	3.8616	{2,2,2,3,3,3,3,3,3}	9	9
10	4.6153	{1,4,5}	3	13	25	3.8420	{2,2,2,2,2,3,3,3,3,3}	10	10
11	4.4571	{2,4,5}	3	11	26	3.7494	{1,2,2,2,2,2,3,3,3,3,3}	11	9
12	4.5079	{4,4,4}	3	15	27	3.7285	{1,2,2,2,2,2,2,2,3,3,3,3}	12	10
13	4.5455	{1,4,4,4}	4	14	28	3.6494	{1,1,2,2,2,2,2,2,2,3,3,3,3}	13	9
14	4.4290	{2,4,4,4}	4	12	29	3.5991	{2,2,2,2,2,2,2,2,2,2,3,3,3}	13	9
15	4.2925	{1,2,4,4,4}	5	11	30	3.6110	{2,2,2,2,2,2,2,2,2,2,2,2,3,3}	14	10

表 8 高计算复杂度下本文方法的编码延迟时间($n=60$)

$n-k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
STCs(ms)	–	–	–	–	–	3.102	3.653	4.172	4.697	5.204	5.661	6.074	6.504	6.895	7.290
Proposed method(ms)	0.001	0.001	0.001	0.001	0.001	0.048	0.041	0.032	0.027	0.228	0.134	0.270	1.407	0.894	2.103
$n-k$	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
STCs (ms)	7.684	8.053	8.384	8.701	9.038	9.352	9.687	9.985	10.29	10.59	10.90	11.17	11.43	11.66	11.89
Proposed method(ms)	4.095	7.133	2.867	2.866	2.441	3.006	5.802	3.443	3.443	7.606	4.216	9.107	4.982	4.981	9.945

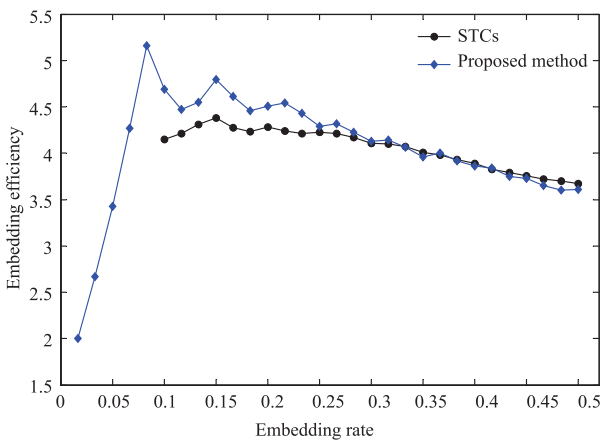


图7 高计算复杂度下本文方法的嵌入效率(n=60)

6 结论

本文分析并阐述了快速矩阵嵌入的基本思想,以此为指导设计了适用于小嵌入率隐写的校验矩阵构造方法. 该方法有效融合了汉明码,与已有矩阵编码相比,进一步提高了嵌入效率和嵌入速度.

参考文献

[1] 师夏阳,马赛兰,胡永健,等. 一种基于像素块的纹理优先自适应隐写算法[J]. 电子学报,2015,43(6):1094-1100.
Shi Xia-yang, Ma Sai-lan, Hu Yong-jian, et al. A pixel block-based adaptive steganographic algorithm with embedding priority given to image textures[J]. Acta Electronica Sinica,2015,43(6):1094-1100. (in Chinese)

[2] Crandall R. Some notes on steganography[EB/OL]. http://www.dia.unisa.it/~ads/corso-security/www/CORSO-0203/steganografia/LINKS%20LOCALI/matrix-encoding.pdf,2015-05-10.

[3] Westfeld A. High capacity despite better steganalysis: F5-a steganographic algorithm[A]. Proceedings of the 4th Inter-

national Workshop on Information Hiding [C]. Berlin: Springer-Verlag, 2001. 2137: 289 – 302.
[4] Fridrich J, Lisonek P, Soukal D. On steganographic embedding efficiency[A]. Proceedings of 8th International Workshop on Information Hiding[C]. Berlin: Springer-Verlag, 2006. 4437: 282 – 296.
[5] Fridrich J, Filler T. Practical methods for minimizing embedding impact in steganography[A]. Proceedings of Security, Steganography, and Watermarking of Multimedia Contents IX, Part of the SPIE-IS&T Electronic Imaging Symposium [C]. Bellingham: SPIE Press, 2007. 6505: 201 – 215.
[6] Filler T, Judas J, Fridrich J. Minimizing embedding impact in steganography using trellis-coded quantization[A]. Proceedings of Media Forensics and Security II, Part of the SPIE-IS&T Electronic Imaging Symposium [C]. Bellingham: SPIE Press, 2010. 7541(1): 175 – 178.
[7] 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.
[8] Fridrich J, Soukal D. Matrix embedding for large payloads [J]. IEEE Transactions on Information Forensics and Security, 2006, 3(1): 390 – 395.
[9] Mao Qian. A fast algorithm for matrix embedding steganography [J]. Digital Signal Processing, 2014, 25: 248 – 254.
[10] Tian Hui, Qin Jie, Huang Yong-feng, et al. Optimal matrix embedding for voice-over-IP steganography [J]. Signal Processing, 2015, 117: 33 – 43.
[11] Wang Chao, Zhang Wei-ming, Liu Jiu-fen, et al. Fast matrix embedding by matrix extending [J]. IEEE Transactions on Information Forensics and Security, 2012, 7(1): 346 – 350.
[12] Gao Yun-Kai, Li Xiao-long, Zeng Tie-yong, et al. Improving embedding efficiency via matrix embedding: a case study[A]. Proceedings of 16th IEEE International Conference on Image Processing [C]. Washington DC: IEEE Computer Society, 2009. 109 – 112.
[13] Wang Jun-jie, Chen Hou-shou. A suboptimal embedding algorithm with low complexity for binary data hiding [A]. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing [C]. Washington DC: IEEE Computer Society, 2012. 1789 – 1792.
[14] Wang Jun-jie, Lin Chi-yuan, Chen Hou-shou, et al. A suboptimal embedding algorithm for binary matrix embedding [A]. Proceedings of International Symposium on Computer, Consumer and Control [C]. Washington DC: IEEE Computer Society, 2012. 165 – 168.
[15] Yan Shu-fan, Tang Guang-ming, Sun Yi-feng, et al. A triple-layer steganography scheme for low bit-rate speech streams [J]. Multimedia Tools & Applications, 2015, 74(24): 11763 – 11782.
[16] Liu Jin, Zhou Ke, Tian Hui. Frame-bitrate-change based steganography for voice-over-IP [J]. Journal of Central South University, 2014, 21(12): 4544 – 4552.
[17] CMU ARCTIC database [DB/OL]. http://www.festvox.org/cmu_arctic/, 2015-07-22.
[18] 黄永峰,李星. IP 语音包的自适应编码和封装算法的研究[J]. 电子与信息学报,2002,24(12): 1829 – 1834.
Huang Yong-feng, Li Xing. An adaptive voice coding and packeting scheme for IP telephony [J]. Journal of Electronics and Information Technology, 2002, 24(12): 1829 – 1834. (in Chinese)
[19] ITU-T, P. 862. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs [EB/OL]. http://www.itu.int/rec/T-REC-P.862/, 2015-06-08.

作者简介

高瞻瞻　男,1988 年生于河北正定. 解放军信息工程大学博士研究生. 主要研究方向为信息隐藏、多媒体信号处理.
E-mail: gaozhandyx@126.com

韦大伟　男,1962 年生于陕西渭南,解放军信息工程大学副教授. 主要研究方向为信息安全、数据挖掘.