一种新的基于双伪随机数的图像隐写算法

周治平[1),2)] 康 辉[2)] 纪志成[2)]

[1)](华东理工大学信息科学与工程学院，上海 200237) [2)](江南大学通信与控制工程学院，无锡 214122)

摘 要 为提高数字图像隐写中的隐藏信息容量，提出了一种基于双伪随机数的图像隐写算法。首先介绍了伪随机数生成和双像素嵌入信息原理，然后将随机产生的整数伪随机数看作一个辅助像素值，结合载体图像中的一个像素值将两位秘密信息同时嵌入到一个载体图像像素中，从而在像素改变较小的情况下提高隐藏容量。最后分析了算法的嵌入性能，并通过仿真实验证明，该方法在提高隐藏信息容量的同时，也具有很好的安全性能。

关键词 图像隐写 伪随机数 最不重要位

中图法分类号：TP391 **文献标识码**：A **文章编号**：1006-8961(2006)10-1405-04

A New Image Steganography Based on Pseudo Random Number

ZHOU Zhi-ping[1),2)], KANG Hui[2)], JI Zhi-cheng[2)]

[1)](College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237)

[2)](College of Communication and Control Engineering, Southern Yangtze University, Wuxi 214122)

Abstract In order to enhance the capacity of steganography, in this paper, an image steganographic algorithm based on Pseudo Random Number(PRN) is proposed. The same Pseudo Random Number sequence is generated by Pseudo Random Number Generator(PRNG) in the parameter certain situation. To utilize this nature of PRNG, the Integer Pseudo Random Number (IPRN) is generated and regarded as an auxiliary pixels value. First, two secret information are hided to one IPRN and one pixel value, and then the secret information which is in the IPRN is displaced to the pixel value. In this way, the two secret information bits are embedded into the pixel value. This method's characteristic is that the capacity of steganography will be enhanced in one time. The experiment proved that this method has a very good security performance and the hiding capacity.

Keywords image steganograph, pseudo random number, LSB(least significant bits)

1 引 言

隐写技术是利用人类感觉器官的不敏感性即感觉冗余，以及多媒体数字信号本身存在的数据特性冗余，将秘密信息隐写到一个宿主信号中而不被觉察。在各种隐写算法中，LSB (least significant bits)嵌入是出现较早的一种隐写算法。LSB 嵌入具有隐藏数据量大，对原始数据的修改小，不易被感官察觉等特点，在实践中被广泛采用。常见的采用 LSB 嵌入的隐写算法有 EZSetgo、Stools、Jsteg, JPHide 和 OutGuess 等[1]，大多数的 LSB 隐写算法实现都和伪随机数(PRN, pseudo random number)结合起来完成信息的嵌入。文献[2]提出了一种基于随机序列填充的信息隐藏方法，该方法使用黄金分割算法计算出被隐藏信息的大小，然后使用填充算法将秘密信息隐藏在随机序列中；文献[3]提出了一种双像素隐写方法(DP_LSB, Double Pixels LSB)。但是简单 LSB 隐写算法抗攻击性能较差，尤其在嵌入较大信息量时。本文在文献[3]的基础上，提出了一种新的采用双伪随机数方法(PRN_LSB)，将待嵌入的秘密信息嵌入到载体图像中，一方面提高了嵌入信息量，另一方面改进了隐写算法的鲁棒性。

收稿日期：2006-04-07；**改回日期**：2006-05-22

第一作者简介：周治平(1962～)，男，副教授。现为华东理工大学博士研究生。主要研究方向为计算机智能控制、信息安全、图像与信号处理等。E-mail：zzping@sytu.edu.cn

2 伪随机数与 LSB 嵌入

2.1 伪随机数 PRN 生成

伪随机数是指用数学递推公式所产生的随机数。它有一个特点就是在参数一定的情况下每次产生的伪随机数序列是相同的。目前常见的伪随机数产生方法有线性同余法[4]、平方取中法、菲波那契(Fibonecci)法和小数开方法。其中线性同余法是应用得较为广泛的一个方法,具有产生速度快、输出序列周期长等特点。线性同余法含有 4 个参数,即

模数 $m(m>0)$;

乘数 $a(0\leqslant a\leqslant m)$;

增量 $c(0\leqslant c\leqslant m)$;

初值即种子 $seed(0\leqslant seed<m)$。

使用迭代公式:$x_{n+1}=(ax_i+c)\bmod m$

得到随机数序列$\{x_n\}$。

2.2 DP_LSB 嵌入原理

如果二值函数$f(x,y)$满足关系式:

$$\begin{cases}f(x-1,y)\neq f(x+1,y)\\ f(x,y)\neq f(x,y+1)\end{cases}\quad \forall x,y\in \mathbf{Z} \tag{1}$$

那么该二值函数中 x 在 ±1 范围内的变化可以使 $f(x,y)$产生不同的值,y 加减 1 同样也会改变$f(x,y)$的值[3]。利用二值函数的这一性质可以将两位秘密信息以较低的像素改变率同时嵌入到两个像素中。实际中可以找到这样的二值函数,如下式所示:

$$f(m,n)=lsb\left(\left\lfloor\frac{m}{2}\right\rfloor+n\right) \tag{2}$$

式中,$m,n\in\mathbf{Z}$,$lsb(\cdot)$表示取二进制像素的最低有效位,$\lfloor\cdot\rfloor$表示向下取整数。以伪代码的形式说明该嵌入原理如下:

```
Input: a pair of cover image pixles(cover one and cover two),
       two message bits(message one and message two);
Output: two stgeo image pixels(stego one and stego two).
if  message one = LSB(cover one)
    if  message two≠f(cover one, cover two)
        stego two ← cover two +1
    else  stego two ← cover two
          stego one ← cover one
    end
else
    if  message two =f(cover one -1, cover two)
        stego one ← cover one -1
    else  stego one ← cover one +1
          stego two ← cover two
    end
end
```

3 基于双伪随机数的隐写方法(PRN_LSB)

为提高信息嵌入容量,在上述嵌入原理的基础上,将两位秘密信息同时嵌入到灰度图像的一个像素中。在嵌入过程中,根据伪随机数发生器在参数一定的情况下每次产生的伪随机数序列相同这一性质,用伪随机数发生器产生的 0~255 之间的整数伪随机数替代上述嵌入算法中的第 2 个载体像素值(*cover two*),将两位秘密信息嵌入到上述算法中第 1 个载体像素值和第 2 个载体像素值(这里为替代后的伪随机数)中,然后再将该伪随机数中的信息转换到图像像素中。

3.1 信息嵌入原理

假设载体图像两个像素灰度值为 x_1,x_2,将要嵌入的秘密信息为 m_1,m_2。则其对每第 2 个载体像素值(*cover two*)的修改概率为

$$P=P(x_1=m_1)P(f(x_1,x_2)\neq m_2) \tag{3}$$

若秘密信息中 0、1 个数近似相等,则有:

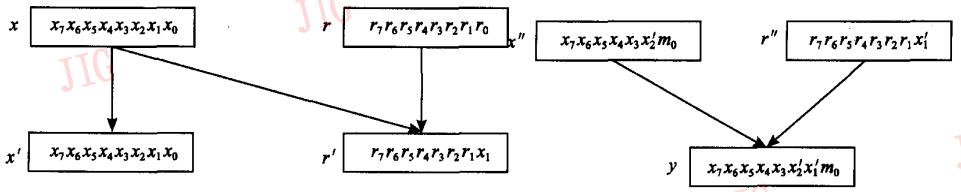
$$P(x_1=m_1)=P(f(x_1,x_2)\neq m_2)=0.5$$

依式(3)得 $P=0.25$。

由此可推论,若将两位秘密信息同时嵌入到一个像素二进制值的最低两位就会很大程度地提高算法的嵌入率,而且对图像本身的改变不大。

嵌入过程中,首先按照图 1(a)对伪随机数产生器产生的伪随机数和载体图像的像素值进行变换,然后按前述算法原理中的两像素嵌入方法将秘密信息嵌入到变换后的伪随机数和载体图像的像素值中,最后按图 1(b)所示得到嵌入秘密信息后的隐藏图像像素灰度值。嵌入算法实现的伪代码如下:

```
Input: one cover image pixel(cover one) and Pseudo Random
       Number(PRN)
       two message bits: message one and message two
Output: one stego image pixel
Change cover to cover_change, PRN to PRN_change according to
Fig. 1(a)
if  message one = LSB(cover_change)
    if  message two≠f(cover_change, PRN_change)
        stego PRN_change ← PRN_change +1
    else  stego PRN_change ← PRN_change
```



(a) 嵌入过程　　(b) 提取过程

图 1　像素与整数伪随机数变换示意图

Fig. 1　The sketch map of switch of pixel and IPRN

stego one ← *cover_change*

else

if　*message two* = *f*(*cover_change* − 1, *PRN_change*)

stego cover_chang ← *cover_change* − 1

else　*stego cover_chang* ← *cover_change* + 1

stego PRN_change ← *PRN_change*

Change *stego cover_change* and *stego PRN_chang* to *stego one* according to Fig. 1(b)

3.2　秘密信息的提取

在秘密信息提取时,首先把产生的伪随机数的 LSB 位用对应隐藏图像像素灰度值的 LSB 相邻位替换,并把该隐藏图像像素灰度值用图 1(a)的方式变换为 7 位二进制数。然后从变换后的伪随机数和像素灰度值中提取出秘密信息。

算法实现的伪代码如下:

Input: one stego image pixel(*stego*) and one Pseudo Random Number(*PRN*)

Output: two message bits(*message one* and *message two*).

Change *stego* to *stego_change*, *PRN* to *PRN_change* according to Fig. 1(a)

message one ← LSB(*stego_change*)

message two ← f (*stego_change*,*PRN_change*)

4　算法性能分析和仿真实验

4.1　算法性能分析

(1)隐藏容量

以 8 位灰度图像为例,PRN_LSB 嵌入是将两位秘密信息同时嵌入到一个像素中,这样嵌入容量就等于载体图像像素个数的 1/4。简单 LSB 嵌入是将秘密信息一位一位嵌入到每个像素中,其嵌入容量为载体图像像素个数的 1/8。而 DB_LSB 嵌入方法则是将两位秘密消息分别嵌入到两个像素中,同样,其嵌入容量只有载体图像像素个数的 1/8。由此可见,3 种隐写算法中 PRN_LSB 嵌入容量为另两种算法的两倍。

(2)嵌入效率

定义载体利用率 R 为嵌入容量 C 与载体像素个数 K 之比,嵌入效率 E 为每个像素的修改概率 P 与该像素可以平均嵌入的比特数 N

$$R = \frac{C}{K} \qquad E = \frac{P}{N} \tag{4}$$

在秘密信息中 0、1 个数近似相等的情况下,可以计算 PRN_LSB 嵌入、DP_LSB 嵌入和简单 LSB 嵌入对每个像素的修改率分别为 3/4、3/8、1/2;载体利用率分别为 200%、100%、100%;嵌入效率分别为 8/3、8/3、2。显然,PRN_LSB 嵌入与 DP_LSB 嵌入效率相同,比简单 LSB 嵌入要高,而载体利用率则提高了 1 倍。

(3)安全性

比较简单 LSB 嵌入和 DP_LSB 嵌入两种隐写算法,PRN_LSB 隐写算法在嵌入过程中采用了双伪随机数的形式,两种伪随机数的可以任意选择使该算法有了更好的加密能力。同样原因,针对秘密信息从隐藏图像中的提取也增加了难度。因此,达到了提高算法抗攻击能力的目的。

4.2　仿真实验

实验取 256×256 的 256 色灰度 bmp 格式图像为样本图片,比较简单 LSB 嵌入、DP_LSB 嵌入和 PRN_LSB 嵌入 3 种隐写算法。算法实现中,用 MD5 函数产生随机无碰撞的像素选择策略控制秘密信息的嵌入位置,用线性同余伪随机数产生器产生伪随机数用做嵌入算法中的辅助像素协助秘密信息的嵌入。

从图像质量方面考虑,用峰值信噪比(PSNR)来测试嵌入信息后图像的质量。用 3 种嵌入方法分别向样本图片 Lena. bmp 中嵌入 800bit、8 000bit、32 000bit、64 000bit、128 000bit 的秘密信息,测试嵌

入后图像的峰值信噪比如表 1 所示。

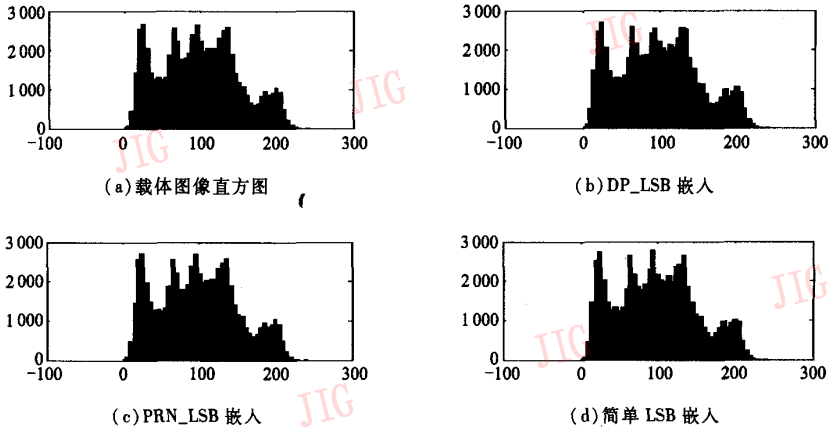
表 1 样本图片 Lena. bmp 峰值信噪比比较

Tab. 1 Comparation of PSNR

实现算法	嵌入量(bit)					
	800	8 000	32 000	64 000	80 000	128 000
PRN_LSB	67.759	57.842	51.708	48.652	47.714	45.652
DP_LSB	68.524	58.572	52.536	49.532	—	—
简单 LSB	67.332	57.386	51.298	48.256	—	—

从表 1 中可以看出 PRN_LSB 嵌入的图像峰值信噪比与 DP_LSB 嵌入的图像峰值信噪比相差不大，略高于简单 LSB 嵌入，且在嵌入信息接近最大嵌入容量时，仍能保持 45.652dB 的峰值信噪比，高于人眼可辨别的峰值信噪比[5]38dB，表明嵌入信息后对载体图像的改变较小，具有较好的隐秘性。

从直方图统计特性方面考虑，PRN_LSB 嵌入对像素灰度值的改变涉及到了 LSB 的相邻位，但是对相邻位改变的概率仅有 0.25。在相同嵌入容量情况下，PRN_LSB 隐写算法对像素的改变个数最小。图 2 是 3 种嵌入算法嵌入后图像的像素灰度值的直方图统计特性，从图中可以看出 3 种隐写算法嵌入信息后对载体图像的直方图都产生了不同程度变化，但这种变化相对来说是比较小的。



(a)载体图像直方图 (b)DP_LSB 嵌入

(c)PRN_LSB 嵌入 (d)简单 LSB 嵌入

图 2 样本图片 Lena. bmp 直方图统计特性

Fig. 2 Statistic characteristic of histogram

5 结 论

通过对伪随机数生成和 LSB 隐写方法的研究，提出了一种利用双伪随机数嵌入秘密信息的隐写算法。该方法用一个伪随机数序列来选择嵌入位置，用另一个伪随机数序列作为辅助像素值协助完成秘密信息的嵌入过程，有效地提高了载体图像的利用率和隐藏信息的容量；同时，由于伪随机数本身的特点和采用双伪随机数进行嵌入，算法安全性能得到了保证；最后，性能分析和仿真实验结果表明了该隐写算法的有效性。

参考文献(References)

1 Neil F Johnson. Steganography Tools[EB/OL]. http://www.jjtc.com/Security/stegtools.htm, 2006-05.

2 Tang Song-sheng, Lü Li-ping. An information hiding method based on filling random sequences [J]. Computer Engineering and Applications, 2005, (30):153 ~ 154. [唐松生,吕丽平. 一种基于随机序列填充的信息隐藏方法[J]. 计算机工程与应用, 2005, (30): 153 ~ 154.]

3 Mielikainen J. LSB matching revisited[J]. IEEE Signal Processing Letters, 2006, 13(5): 285 ~ 287.

4 Yang B. Modern Cryptology [M]. Beijing: Qsinghua University Publishing House, 2003: 130 ~ 131. [杨波. 现代密码学[M]. 北京:清华大学出版社, 2003: 130 ~ 131.]

5 Petitcolas F A P, Anderson R J. Evaluation of copyright marking systems[A]. In: Proceedings of IEEE Mulitimedia Systems[C], Florence., Italy, 1999: 574 ~ 579.