# 一种基于 LSB 的数字图像信息隐藏算法*)

刘红翼 王继军 韦月琼 苏 勤 黄 涛 张显全
(广西师范大学计算机科学系 桂林 541004)

**摘 要** 根据载体图像和嵌入量的大小,利用随机函数确定出嵌入字节的位置,使嵌入信息分布均匀;研究了异或运算的性质,结合位平面的特点,通过位的异或运算,实现秘密信息嵌入;对嵌入的字节最多只需修改一位,可在该字节中同时嵌入两位秘密信息,提高了秘密信息嵌入量,并能无损还原。理论分析和实验结果证明了算法的有效性。

**关键词** 最低有效位,信息隐藏,位平面

## An Algorithm for Digital Image Information Hiding Based on LSB

LIU Hong-Yi WANG Ji-Jun WEI Yue-Qiong SU Qin HUANG Tao ZHANG Xian-Quan
(Department of Computer Science, Guangxi Normal University, Guilin 541004)

**Abstract** This paper presents a new method to ascertain the position of embedding bytes according to the size of cover image, embedding capacity and random function. The embedding information is distributed evenly. The secret information is hiding though the properties of XOR operation and the features of bit planes. The algorithm makes the two secret information bits embed into a pixel, but only need change one bit at most. It can enhance the embedding capacity and the secret information can be recovered with no change. The experimental results show that this method is available and efficient.

**Keywords** LSB(least significant bit), Information hiding, Bit planes

## 1 引言

近几年来,随着 Internet 的迅速发展和广泛应用,信息安全问题日益突出,信息隐藏技术[1]作为一种新兴技术,已成为信息安全领域的一个重要组成部分。信息隐藏是在网络环境下把机密信息隐藏在其它无关紧要的信息中形成隐秘信道,除通信双方以外的任何第三方并不知道包含秘密通信这个事实的存在。与采用密码加密方法的保密通信相比,信息加密从"看不懂"变为"看不见",转移了攻击者的目标,这种技术具有良好的透明性,即使你看见了、听见了还是不能有所发现,并且加入秘密信息前后,媒体在网络中的传输没有什么不同。信息隐藏技术包括水印技术和隐写技术,水印技术在防伪和版权保护中有着广泛应用,而隐写技术起源较早,现在被更多地应用于安全通信、电子政务、电子选举、电子商务和数字图书馆等领域。同时人们也越来越关注如何察觉、控制隐蔽通信的问题,在隐写攻击等方面做了大量的研究工作。

目前,信息隐藏方法各式各样,其中最低有效位算法 LSB (least significant bit)因其实现简单,嵌入量大而被广泛应用,且大部分的多媒体文件(如图像、音频和视频文件等)都可作为 LSB 算法的载体。文[2,3]中最早提出针对 LSB 替换信息伪装的隐写算法,通过分析像素值统计分布建立卡方统计量来检测隐藏信息的存在,并能计算出嵌入秘密信息的大小;文[4]给出了一种新的图像隐藏方案,使得隐藏秘密信息后的图像既包含隐藏信息又能包含密钥信息;文[5]将混沌序列作为一种噪声的扩谱序列,把所要传送的敏感信号以白噪声的扩谱信号调制隐藏在载体图像中进行安全通信;文[6]在分析图像像素灰度差分的基础上,提出一种自适应信息隐藏算法,它充分利用人类视觉特性,结合图像的局部特征自适应地嵌入秘密信息,使得对载体图像的改变具有很好的不可见性。文[7]采用信息论的观点对信息隐藏系统做了分析,给出了用香农熵和相对熵来定义安全隐藏系统的方法。

本文研究了异或运算的性质,并将其用于信息隐藏,算法首先根据嵌入量的大小确定嵌入位置,使秘密信息尽可能分散,然后通过秘密信息与待嵌入信息进行比较,确定是否需要修改位,实现秘密信息的嵌入,增强了隐蔽性;通过位运算,可在一字节中嵌入两位秘密信息,提高了嵌入量,理论和实验结果均表明,本文的算法简洁、嵌入量大,隐蔽性好,并能无损还原,有一定的应用价值。

## 2 LSB 图像信息隐藏算法

对于信息的隐藏,关键是嵌入位置和嵌入方式的确定,要确保嵌入秘密信息后载体图像没有明显的视觉变化,统计特性也没有明显差异,同时嵌入后的图像要有较强的抗功击性,嵌入和提取方法要有较高的效率。

### 2.1 嵌入位置的确定

一幅数字图像可以用一个二维矩阵来表示,矩阵的各个元素代表一个像素的色彩信息,矩阵中的元素数值可以是双精度、无符号 8 位整型(unit8)等,而在信息隐藏中常用到的是 unit8。同时根据图像采用的颜色数,可以将图像分为 2 位、8 位和 24 位图,并将图像中每个像素点颜色值的某一位共同构成的一个新的二值图像称为该图像的一个位平面图像。一般定义从图像的第 0 个位平面到第 7 个位平面依次为最不重要位平面到最重要位平面,相应的位被称为最低有效位 LSB(Least Significant bits)和最高有效位 MSB(Most Sig-

*)基金项目:广西自然科学基金(0447035);广西教育厅项目;广西研究生教育创新计划项目(2006106020812M37)。刘红翼 讲师,主要研究方向:图形图像处理;王继军 硕士研究生,主要研究方向为图像处理,信息隐藏;韦月琼、苏 勤、黄 涛 硕士研究生。

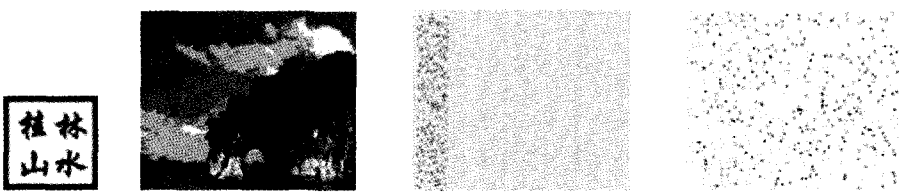
nificant bits)，在信息嵌入时，修改不同的位对图像的影响不同，图1给出的是依次将标准Lena图中每个像素各分量的第0到7位清0后的图像，它们分别是第0到7个位平面的补图，反映了每个位平面对图像的影响，从实验结果可以得出，MSB对图像影响最大，修改MSB(第七个位平面)后，图像的色彩已经完全被破坏，而第0、1和2三个位平面的改变对图像的影响小，且位平面越低，对图像的影响越小，因此可通过修改低位平面上的位值，实现图像中秘密信息的隐藏。

(a)原图 (b)位平面0补图 (c)位平面1补图 (d)位平面2补图 (e)位平面3补图 (f)位平面4补图 (g)5位平面5补图 (h)位平面6补图 (i)位平面7补图

图1 位平面补图

在已有的LSB图像信息隐藏算法中，大多数隐藏是对最低位进行操作，并以连续顺序嵌入方式居多，但单一的连续顺序嵌入方式可能使秘密信息高度集中在载体图像的某个区域，造成载体图像各部分统计特性的不一致，增大了攻击者对秘密通信的怀疑，图2(c)就是图2(a)连续顺序嵌入图2(b)的位置图，嵌入的秘密信息过于集中，可能对统计特性有较大影响。为了使秘密信息能均匀分布在载体图像内，减少视觉变化和统计特性的变化，在本文方中使用如下的方法确定嵌入字节。实际上就是确定要在载体图像的哪一个像素点上嵌入秘密信息，设载体图像的容量为 $L$ 个字节，秘密信息的长度为 $l$ 位，$\alpha$ 为每个字节中欲嵌入秘密信息的总位数，$t=(L*\alpha)/l$，若能实现嵌入，则必然有 $L/l>1$，而 $\alpha\geqslant 1$，所以 $t\geqslant 1$，则在长为 $t$ 字节的跨度间至少有一个字节可嵌入秘密信息，在每个 $t$ 跨度内随机选择一个字节作为嵌入字节，实现均匀随机嵌入。首先以密钥产生随机序列 $k[0],k[1],\cdots,k[l-1]$，设第 $i$ 个选择插入的字节位置为 byte$[i]$(在图像中应为整数)，则该字节在 int$((i-1)t)$ 和 int$(i\times t)$ 之间，由 $k[i]$ 来生成嵌入字节位置为：

$$\text{byte}[i]=\text{int}(i\times t)+k[i] \bmod (\text{int}((i+1)t)-\text{int}(i\times t))(0<i\leqslant l)$$



(a)隐藏信息 (b)载体图像 (c) 顺序连续嵌的位置 (e)本文算法的的位

图2 信息隐藏位置对比图

上述方式既实现了随机选择，又避免了重复和越界，也实现了均匀分布，当 $t=1$ 时，所有字节都被选中，这时成为顺序连续嵌入方式，通过以上方式实现了选择嵌入字节的位置。图2(e)就是图2(a)用本文算法嵌入图2(b)的位置图，与图(c)中嵌入的信息完全相同，但嵌入信息分布较为均匀，同时也由于采用了随机选择嵌入方式，进一步增加了攻击的复杂度，这种嵌入方法有一定的优越性。

### 2.2 嵌入算法

秘密信息的嵌入是一个信息隐藏的过程，提取运算是嵌入的逆运算，也是秘密信息的恢复过程。对于秘密信息的嵌入过程，首先选择载体图像嵌入的像素点，然后直接或经某种运算在这些像素点上执行替换或修改操作；对于提取过程，首先要找出载体图像中隐藏时使用的像素序列，然后提取信息，最后按嵌入时顺序排列，便可重构出秘密信息。

在LSB信息隐藏算法中，因图像的最低位对人的视觉影响最小，使得很多嵌入都是对最低位操作，但若所有的操作都集中到LSB平面上，很容易引起攻击者的怀疑。在对秘密信息分析方法研究中，很多是针对最低位着手的(如：LSB统计特性等)，因此在信息嵌入时应尽可能地避免只对最低位操作，本文结合秘密信息的分散程度、嵌入量大小和隐蔽性好坏等因素，应用位运算的性质，提出了一种新的信息隐藏方法。对位运算有如下性质：

**性质1** 设 $x_i\in\{0,1\}$，$T=x_0\oplus x_1\oplus,\cdots,\oplus x_n$，$m=\sum_{i=0}^{n}x_i$，则当 $m$ 为偶数时，$T=0$，$m$ 为奇数时，$T=1$。

证明(略)。由性质1可看出：$T$ 的值只与 $m$ 的奇偶有关，而与 $x_i(i=0,1,\cdots,n)$ 的排列次序无关。

**性质2** 设 $x_i\in\{0,1\}$，$T=x_0\oplus x_1\oplus\cdots\oplus x_{i-1}\oplus x_i\oplus x_{i+1}\cdots\oplus x_n(0\leqslant i\leqslant n)$，如果将其中某一个 $x_i(0\leqslant i\leqslant n)$ 位 $\bar{x}_i$ 用替换，则 $x_0\oplus x_1\oplus\cdots\oplus x_{i-1}\oplus\bar{x}_i\oplus x_{i+1}\cdots\oplus x_n=\bar{T}$。

证明(略)。对于嵌入位的确定，实际上就是要确定把密秘信息嵌入被选中字节的哪一位中，与嵌入字节的确定方法类似，由位平面分析的结果可知，修改低三位的值，对图的影响较小，因此嵌入位一般只在低三位中选择。下面对每个字节修改1位，嵌入1位和2位进行讨论，即 $\alpha=1$ 和 $\alpha=2$ 两种情况：

当 $\alpha=1$ 时，可以仍采用上面以密钥 $k$ 生成随机序列 $k[i]$，只对低三位进行修改，构造位选择器 bit$[i]$：

$$\text{bit}[i]=k[i] \bmod 3(0\leqslant i<l)$$

那么通过确定嵌入位和嵌入的字节，对应得到的二元组(byte$[i]$,bit$[i]$)就唯一确定了要嵌入的字节以及该字节上要嵌入的位，它本身具有随机性。设载体图像的某一字节的后三位分别为 $x_2$、$x_1$ 和 $x_0$，$m=\sum_{i=0}^{2}x_i$，$s$ 为秘密信息位，在信息嵌入时，若嵌入位 $s=0$，根据性质1和2可修改低三位中任一位或不修改，使 $m$ 为偶数，若嵌入位 $s=1$，同样根据性质1和2可修改低三位中的任一位或不修改，使 $m$ 为奇数；这样可根据 $m$ 的值对嵌入信息进行恢复。对 $x_1$、$x_2$ 和 $x_0$ 的处理情

况，如表1所示。

表1 嵌入1位的位处理情况

| $s$ | $M$ | 嵌入位处理 | 结论 |
|---|---|---|---|
| 0 | 偶数 | 不需修改 | 若低三位和为偶数，提取的值为0，为奇数提取的值为1。 |
| | 奇数 | 对 $x_0$、$x_1$ 和 $x_2$ 中某位取非 | |
| 1 | 偶数 | 对 $x_0$、$x_1$ 和 $x_2$ 中某位取非 | |
| | 奇数 | 不需修改 | |

例如：被选中的某字节的后三位 $x_2$、$x_1$ 和 $x_0$ 为：101，很显然后三位中1的个数 $m=2$ 是偶数，此时，如果欲嵌入信息位 $s=0$，则(2 mod 2)⊕0=0，无需任何改变，提取时 $m'=2$ 偶数，$x=0$；若 $s=1$，(2 mod 2)⊕1=1，则需在101中选一位将其取非，不妨设选中的是 $x_2$，则嵌入后的后三位变为111，提取时因 $m'=3$ 为奇数，所以 $s=1$。

上述方法虽然实现了嵌入，效果也较为理想，但修改一位只能嵌入了一位信息，对嵌入量有局限性，但同一字节中同时修改多位对图像影响较大，隐蔽性较差，本文提出如下方法，能在修改一位时嵌入两位秘密信息，算法在修改总位数相同的前提下，嵌入量提高了一倍。

当 $\alpha=2$ 时，假定欲在某选定字节的低3位 $x_2$、$x_1$ 和 $x_0$ 上嵌入2个秘密信息位 $s_0$ 和 $s_1$，设 $m_0=x_0\oplus x_2$，$m_1=x_1\oplus x_2$，可通过在 $m_0$ 与 $s_0$、$m_1$ 与 $s_1$ 之间分别建立一种关系实现嵌入，若秘密信息嵌入后，关系式 $\begin{cases}m_0=s_0\\m_1=s_1\end{cases}$ 成立，那么提取时，只需计算出每一个被嵌入字节中的 $m_0$ 和 $m_1$，将 $m_0$ 和 $m_1$ 的值合理排序后，便可重构出秘密信息，为了确保上述关系式始终成立，对嵌入的不同条件，根据性质2，对 $x_2$、$x_1$ 和 $x_0$ 做相应修改，如表2所示。

表2 字节中嵌入2位的位处理情况

| 初始条件 | 需修改位 | 结论 |
|---|---|---|
| $s_0=m_0$ 且 $s_1=m_1$ | 无 | 初始条件包含了所有可能情形，并在确保最多只修改一位的前提下，实现了两位的嵌入且 $m_0$ 和 $m_1$ 的值恰好是嵌入信息 $s_0$ 和 $s_1$，嵌入和提取均较为简洁。 |
| $s_0\neq m_0$ 且 $s_1=m_1$ | $x_0$ 取非 | |
| $s_0=m_0$ 且 $s_1\neq m_1$ | $x_1$ 取非 | |
| $s_0\neq m_0$ 且 $s_1\neq m_1$ | $x_2$ 取非 | |

例如：$x_2$，$x_1$，$x_0$ 三位为101，则 $m_0=0$，$m_1=1$，如果待嵌入信息 $s_0=1$，$s_1=1$，则初始条件为 $s_0\neq m_0$ 且 $s_1=m_1$，此时须对取非，取非后的 $x_2$，$x_1$，$x_0$ 变为001。提取时，$m_0=0\oplus 1=1=s_0$，$m_1=0\oplus 1=1=s_1$，对其它情形也均成立。

由上可知，本文提出的算法方法简洁，运算量小，嵌入量大。$\alpha=1$ 时，嵌入一位最多修改一位，并且嵌入字节和嵌入位的确定均有较好随机性，嵌入的秘密信息也较为分散，隐蔽性较好；$\alpha=2$ 时，算法也有较好的随机性，嵌入信息也较为分散，最多只修改1位，能在一个字节中嵌入两位秘密信息，虽然修改总位数的最大值相同，但嵌入量增加了一倍，算法较为理想。

### 2.3 算法实现

本文提出的算法简单，在此仅以 $\alpha=2$ 的算法为例进行说明。

设 $x_7$，$x_6$，$x_5$，$x_4$，$x_3$，$x_2$，$x_1$，$x_0$ 表示载体图像某一像素点的颜色值所对应的8个二进制位，$L$ 为载体图像的总容量，$l$ 为秘密信息的长度，$m_0=x_0\oplus x_2$，$m_1=x_1\oplus x_2$，$s_i\in\{0,1\}(0\leqslant i\leqslant l-1)$，为秘密信息二进制流中的某一位。

step 1：读入载体图像；

step 2：读入秘密信息 *message*，将秘密信息转化为二进制流 $s_i(0\leqslant i\leqslant l-1)$，$l$ 为 *message* 的长度，$t=2L/l$，$p$ 是已嵌入秘密信息位个数，初值为0；

step 3：if($l<L$ && $p\leqslant l$)

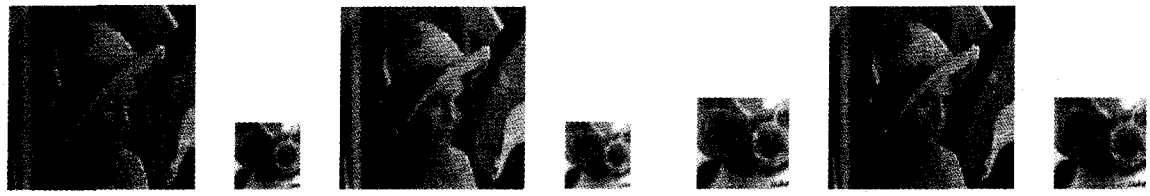
```
for(i=1;i<l;i++)
{
    以密钥 K 产生一个随机序列 k[i];
    byte[i]=int(i×t)+k[i] mod (int((i+1)t)-int(i×t));
    获取当前被选中像素 byte[i]的颜色值;
    if(s_i==m_0 && s_{i+1}==m_1)
            p++;
    else if(s_i=m_0 && s_{i+1}==m_1)
            {x_0=¬x_0;p++;}
    else if(s_i==m_0 && s_{s+1}!=m_1)
            {x_1=¬x_1;p++;};
    else if(s_i!=m_0 && s_{i+1}!=m_1)
            {x_2=¬x_2;p++;};
}
```

step4：修改后的图像就是隐藏秘密信息后的图像。

提取是嵌入的逆过程，提取时，只需确定出嵌入字节，并计算机出该字节中相应的 $m_0$ 和 $m_1$，$m_0$ 和 $m_1$ 便是秘密信息的位，当计算出所有位后，合理排序，便可重构原秘密信息。

## 3 实验结果

用本文算法进行了大量实验，对于秘密信息为图像、.txt、.rar等类型的文件均取得了较好的效果，限于篇幅有限，在此仅给出 $\alpha=1$ 和 $\alpha=2$ 的一组实验结果，实验结果如图3所示(图像同比缩小了约40%)，载体图像(a)是大小为256×256像素的标准lena图，秘密图像(b)的大小为90×90，采用的是一个字节中嵌入一位秘密信息的方法，即 $\alpha=1$，嵌入后的图像如图(c)；秘密图像(e)的大小为125×125，采用的是一个字节中同时嵌入两位秘密信息的方法，即 $\alpha=2$，嵌入后的图像如图(f)，从实验结果可以看出，本文隐藏算法有较好的效果。



(a)载体图像 (b)秘密图像 (c)隐藏图a的结果 (d)图c还原图b (e)秘密图像 (f)隐藏图e后的结果 (g)图f还原图e

图3 本文算法隐藏效果图

(下转第125页)

显。如何对待选中间视图预先进行过滤以缩减考察空间，将作为我们进一步的研究方向。此外，算法对非均衡数据源的更新处理也有待进一步研究。

## 参 考 文 献


1 Gupta A, Mumick I. Maintenance of materialized views: problems, techniques, and applications. IEEE Data Eng Bull, 1995, 18(2): 3～19

2 陈长清，程恳，冯玉才. 一种用于搜索可能响应查询的候选实化视图的索引. 计算机科学，2003，30(5)：176～179

3 Salem K, Beyer K S, et al. How to roll a join: asynchronous incremental view maintenance. In: Proceedings of SIGMOD, 2000. 129～140

4 Lee K Y, Son J H, et al. Efficient incremental view maintenance in data warehouse. In: Proceeding of CIKM, 2001. 249～356

5 Liu B, Rundensteiner E A, Finkel D. Maintaining large update batches by restructuring and grouping. Information Systems, 2007 (Accepted)

6 Lee K Y, Kim M H. Optimizing the incremental maintenance of multiple join views. In: the Proceeding of DOLAP, 2005. 107～113

7 王新军，洪晓光，等. 数据仓库中多数据源物化视图的一种有效更新算法. 计算机研究与发展，2004，41(5)：874～879

8 Labio W J, Yerneni R, et al. Shrinking the warehouse update window. In: Proceedings of SIGMOD. June 1999. 383～395

9 DeHaan D, Larson P-A, et al. Stacked indexed views in Microsoft SQL server. In: Proc of SIGMOD, 2005. 179～190

10 Folkert N, Gupta A, et al. Optimizing refresh of a set of materialized views. In: the Proceeding of 31st VLDB, 2005. 1043～1054

11 Skowron A, Rauszer C. The discernibility matrics and function in information system. In: Slowinski R, ed. Intelligent Decision Support Handbook of Application and Advances of the Rough Sets Theory. Dordreecht: Kluwer Academic Publishers, 1991. 331～362

12 TPC BENCHMARKTM H Standard Specification (Decision Support), Revision 2.3.0. http://www.tpc.org/tpch


(上接第 89 页)


20 Frantzen M, Shuey M. StackGhost: Hardware Facilitated Stack Protection. In: Proceedings of the 10 th USENIX Security Symposium, August 2001

21 Shao Z, Cue Chun, Zhuge Qingfeng, H M Sha E. Security Protection and Checking in Embedded System Integration Against Buffer Overflow Attacks. In: Proceeding of the International Conference on Information Technology: Coding and Computing, 2004

22 StackOFFence: a technique for defending against buffer overflow attacks. Madan, B B, Phoha S, Trivedi K S, et al. Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on Volume 1, April 2005. 656～661

23 Lee R B, Karig D K, McGregor J P, et al. enlisting hardware architecture to thwart malicious code injection. Lecture Notes in Computer Science, 2004, 2802: 237～252

24 Paulson L D. 同 New chips stop buffer overflow attacks. Computer, 2004, 37(10): 28

25 Baratloo A, Singh N, Tsai T. Transparent Run-time Defense against Stack Smashing Attacks. In: Proc. of the 9th USENIX Security Symposium, June 2000

26 Baratloo A, Singh N, Tsai T. Transparent Run-Time Defense Against Stack SmashingAttacks. In: Proceedings of the 2000 USENIX Annual Technical Conference, San Diego, California, USA, June 2000

27 Nebenzahl D, Sagiv M, Wool A. Install-Time Vaccination of Windows Executables to Defend against Stack Smashing Attacks. Dependable and Secure Computing, IEEE Transactions on, 2006, 3(1): 78～90

28 Shao zili, Xue Chun, et al. Efficient array and pointer bound checking against buffer overflow attacks via hardware/software. In: International Conference on Information Technology: Coding and Computing, ITCC, v1. Proceedings ITCC 2005 - Intenational Conference on Information Technology: Coding and Computing, 2005. 780～785

29 Hornof L, Jim T. Certifying Compilation and Run-time Code Generation. In: Proceedings of the ACM Conference on Partial Evaluation and Semantics-Based Program Manipulation, January 1999

30 Karger P A, Schell R R. Thirty Years Later: Lessons from the Multics Security Evaluation. In: Proceedings of the 2002 Annual Computer Security Applications Conference, December 2002. 119～126

31 Tan Yu-an Tan, Zheng Ji-yan Zheng, Cao Yuan-da, Zhang Xue-lan. Buffer overflow protection based on adjusting code segment limit. Communications and Information Technology, 2005. ISCIT 2005. In: IEEE International Symposium on Volume 2, 2005. 947～950


(上接第 102 页)

载体图像(a)的大小为 256×256，如果 α=1，修改一位只嵌入一比特位信息，则可嵌入的最大信息量为 256×256 / 8 =8192byte，实验中实际嵌入的秘密信息图(b)大小为 90×90 =8100byte，若 α=2，则可在对载体图像修改相同位数的条件下，增加一倍的嵌入量，从实验结果可看出，单靠人眼视觉很难发现载体图像中藏有秘密信息，两种算法的隐蔽性均较好，并且从上面的分析也可知本文算法对常规的密写分析方法有较强的抵抗能力，算法是有效的、可行的。

## 4 分析与讨论

LSB 信息隐藏是一种基于空域的隐藏算法，空域隐藏算法采用直接改变图像元素值的方法，一般是在图像元素的亮度或色度中加入隐藏的内容，空域类算法的特点是只需对隐秘载体进行很小的、不易察觉的改变就能隐藏很大的信息量，并且信息嵌入和提取算法简单，待隐藏信息格式又可以是任意的文件，因此计算速度较快。同时人眼对绿色最敏感，红色次之，蓝色最不敏感。因此对于载体图像为 24 位位图的图像，可在人眼不敏感的颜色位中嵌入较多比特位，而在敏感的颜色位中嵌入较少比特位，在不影响视觉感观的条件下，增加嵌入量。但其存在的主要局限是对信道干扰及数据操作的抵抗力较弱，当载体文件遭到破坏或受到某些程度噪声的干扰时，嵌入的隐藏数据将很难完整地提取出来。不过，作为一种大数据量的信息隐藏方法，LSB 算法在隐藏通信中仍占据着重要的地位，他也是其它隐藏方法的基础。

另外，目前已有很多变换域隐藏算法，变换域隐藏算法是利用某种数学变换，将图像用变换域(如频域)表示，通过更改图像的某些变换域系数加入待隐藏信息，然后再利用反变换来生成原始图像。常见的变换域算法有：基于 DCT 的变换域算法、基于 DWT 的变换域算法等，这些都是研究的热点。

## 参 考 文 献


1 Aucsmith D. Information hiding. In: Proceedings of the Second International Workshop, Lecture Notes in Computer Science 1525. Berlin: Springer-V erlag, 1998

2 Westfeld A, Pfitzmann A. Attacks on steganographic systems. In: Pfitzmann A, ed. Proc. of the 3rd Int'l. Workshop on Information Hiding. Lecture Notes on Computer Science 1768, Berlin: Springer-Verlag, 1999. 61～76

3 Chang C C, Lin M H, Hu Y C. A Fast and Secure Image Hiding Scheme Based on LSB Substitution [J]. International Journal of Pattern Recognition and Artificial Intelligence, 2002, 16(4): 399～416

4 王道顺，齐东旭. 一种新的数字图像隐藏方案[J]. 计算机学报，2000，23(9)：949～952

5 刘年生，郭东辉. 基于混沌加密的一种图像信息隐藏传送方法[J]. 计算机工程，2006，32(7)：135～137

6 王君，田玉敏，李春霞. 一种改进的图像自适应信息隐藏算法[J]. 计算机应用研究，2005，5(3)：145～147

7 吕欣，马智，冯登国. 安全隐写系统的信息理论分析[J]. 计算机科学，2006，33(6)：140～142