

隐写术（盲水印）：从入门到出门

原创

[guofei9987](#) 于 2020-11-20 01:06:43 发布 5964 收藏 53

分类专栏：[盲水印](#) [数字](#) [图像](#)

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/guofei9987/article/details/109832555>

版权



[盲水印](#) 同时被 3 个专栏收录

3 篇文章 4 订阅

订阅专栏



[数字](#)

2 篇文章 0 订阅

订阅专栏



[图像](#)

2 篇文章 0 订阅

订阅专栏

0. 前言

Stars 1.8k

我在做 [Blind Watermark](#) 这个库的时候，翻阅了大量材料，学到了关于隐写术、盲水印的很多知识，现在梳理了一遍，发出来。

本文结构：

1. 简介：隐写术的应用场景、分类、特点
2. 隐写术：介绍几种巧妙的隐写术的算法
3. 隐写术的衡量指标：如何衡量算法的好坏
4. 隐写分析：如何攻击隐写算法

1. 隐写术的简介

应用场景：

1. 版权识别。数字水印可以提供所有权证据。
2. 用户识别或指纹。合法用户的身份嵌入水印，用于识别非法复制。
3. 保证图像不被篡改。如果水印设计成，对图像任何修改将破坏水印。
4. 自动监视。

从算法效果分类

可见水印：

$$f = \alpha \cdot f + \dots$$

（可见水印不是本主题的重点，下面就不涉及了）

易碎不可见水印：主要应用场景是识别图像在某个传输环节被篡改，用来做图像认证
一个例子是LSB水印

鲁棒水印：对以下攻击鲁棒

- 无意攻击：有损压缩、线性和非线性滤波、修剪、旋转、重取样
- 故意攻击：打印、重扫描、添加水印、添加噪声

鲁棒水印的应用非常广，例如版权保护、盗版跟踪

从加密方式分类

算法分为私钥系统、公钥系统

从解密方式分类

算法分为需要原件才能解密、不需要原件就可以解密

历史上的隐写术

- 不可见墨水：被万用显影剂破解了。万用显影剂可以检测纤维表面，确定那一部分被加湿过。
- 纸币防伪技术，紫外线荧光。
- 微缩胶片，粘到无关杂志的标点符号上。
- 用正/斜体编码，把二进制编码隐藏在印刷文字里。

从嵌入的通道分类

- 替换：用秘密信息替换载体的冗余部分，典型算法是 LSB（下文详细介绍）
- 变换域：例如把密文信息写入频域，典型算法是基于 DCT 的隐写术（下文详细介绍）
- 扩展频谱技术
- 统计方法：更改载体的若干统计特征，在提取时使用假设检验。
 - 用来嵌入1个比特，找到某个统计特征，如果要嵌入1，就显著改变统计特征；如果要嵌入0，就不修改。
 - 解码过程就是一个假设检验函数。
 - 此方法通常应用场景很窄
- 失真技术：使信号失真来保存信息，解码时测量与原始载体的偏差
 - 例子1:在 html 中插入不可见的字符，这些字符在web浏览器中被忽略，但查看 html 源码可以发现
 - 例子2:在印刷品中，单词（或句子，或整行），做微小的上下移动，可以用来嵌入比特。
 - 例子3:在印刷物中，单词之前的空格，也可以有大小，用来嵌入比特。
- 载体生成：直接把信息编码成伪装载体
 - 算法1:直接做文本/字母的替换等，如“与佛说禅”
 - 把信息隐藏在字母出现的频率（Huffman树上）。步骤是：首先根据秘文生成 Huffman 树，然后根据 Huffman 树生成一段文字。

2. 几个隐写术算法

2.1 LBS

LSB（最低有效位，Least Significant Bit）利用人类视觉的一个特点，即对在某一阈值之下的变化感知不明显。

LSB算法比较简单，就是把载体的二进制低位替换为密文。

用公式写，就是用无符号整数运算执行下面的运算：

$$f = 4(f/4) + m/64$$

- 核心原理是，对于8比特图像，最低阶比特对我们感知图像是没有效果的。
- $4(f/4)$ 是把最低的两阶置零。 $w/64$ 把水印压缩到最低两阶
- 算法的特点是易碎，经过有损压缩后，水印全部丢失。

LBS 的优缺点：

- 人眼无法察觉图像变化
- 易受剪切和压缩的影响
- bmp或gif转jpg后，密文就会丢失
- 可嵌入的信息极多。例如，一幅图像是 $800 \times 600 \times 3$ 的8位图，如果只嵌入最低位，可以嵌入 $800 \times 600 \times 3 = 1.44\text{e}6$ bit 数据=180k byte

LSB改进：Hide and Seek

上面提过，LSB可以嵌入极多的数据，那么我们可以放弃一定的嵌入容量，来换取隐蔽性。

Hide and Seek 的思路就是随机选取一些像素点，在这些像素点上嵌入信息。

具体算法就不写了，几个整除。而且方法变种可以有很多。

LSB改进2:LBS替换

规则如下：

- 如果载体最后一位与密文一致，保持最后一位不变
- 如果不一致，随机+1或-1

然后真值表：

原载体最后一位	密文	替换方法	替换后的载体最后一位
1	1	1	1
1	0	± 1	0
0	1	± 1	1
0	0	0	0

攻击LSB

chi2测试

- 假设：计算整体的奇偶比例，假设每个部分的奇偶比例等于整体的
- 如果隐写术顺序插入的，顺序检测到chi2的p值接近1，密文结束时，p值突然变成0。因此不但能检测出是否嵌入密文，还能检测出密文的长度。
- 如果隐写术不是顺序嵌入而是随机嵌入（Hide and Seek），Chi2 几乎失效。
- 因此，要对付 chi2测试，隐写算法必须控制密文信息量，如果密文信息量远小于载体容量，就很难用chi2测试发现隐写了。

拉普拉斯滤波

- 正常的图像，拉普拉斯滤波后的分布图主要在0附近
- 但是 LSB 后的图，拉普拉斯滤波后的柱状图有较大变化，中心区域会变形。这是因为一般图像的像素附近点差别较小，但 LSB 会让附近点有变化。

抹除隐藏的信息：LSB嵌入的隐藏信息很容易抹除

- 压缩图片，例如32位压缩到24位
- JPEG压缩

2.2 频域隐写技术

变换域技术包括：

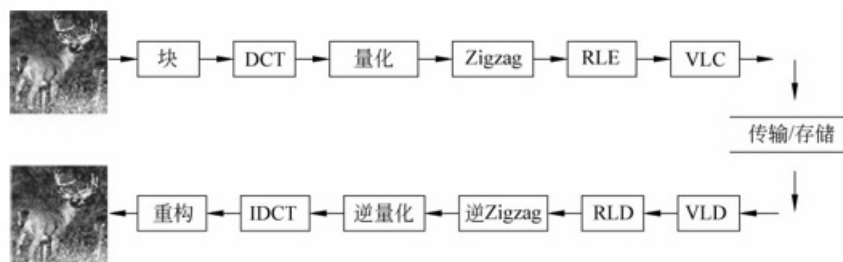
- 离散余弦变换（discretecosinetransform, DCT），DCT变换的好处是，如果原序列是实数序列，那么变换后也是实数序列。二维DCT的最著名的应用是 JPEG 压缩。
- 离散小波变换（discretewavelettransform, DWT）
- 离散傅立叶变换（discretefouriertransform, DFT）
- Mellin傅立叶变换（mellinfouriertransform）

几种主流算法：

- JSteg是第一个有效公开的基于JPEG的隐写程序。
- F5所使用的嵌入操作与LSB匹配不同，它使用矩阵嵌入来保证嵌入的改变最小化。
- MBS1和MBS2使用LSB匹配，并保留DCT系数的模型。
- OutGuess和Steghide保留了一阶统计，Steghide改变系数对。除了OutGuess和F5，所有算法都以JPEG图像为输入，直接调制它们的DCT系数。

JPEG压缩算法

原图的某个通道（所以可以处理RGB或YUV等），做以下处理：



JPEG压缩的流程：

1. 把图片转成 YCbCr 颜色空间
2. 按照 8x8 分块，对每个分块做DCT
3. 根据压缩因子，建立一个定义好的 8x8 的整数矩阵
4. 对第2步的每个分块进行操作，2的结果关于3的整数取整到最接近整数
5. 4的效果是，每个分块 DCT 的每个值的可能性就大大减少了
6. 用 Huffman 编码存下来（或者其他熵编码器）

DCT 隐写算法

1. 伪随机拿出一个8x8分块，这个分块之后用来嵌入一个比特
2. 对分块做 DCT
3. DCT后的结果，伪随机选择2个数字，b1,b2
4. 如果要嵌入的比特值是0，确保 b1 小于 b2；如果要嵌入的比特值是1，确保 b2 小于 b1。如果不满足，用交换的办法来满足。
5. 加入随机性，使 $|b1 - b2| > x$ ，这里面的x越大，算法抵抗 JPEG 压缩的能力越强，但是图像的质量就越差
6. 逆向DCT变化 x ，回到空域

2.3 基于调色板的隐写术

GIF和BMP的调色板索引是可以随意交换顺序的，如果有N种颜色，就可以有 N! 种组合，因此可以用调色板的顺序来埋入信息。

- 鲁棒性很差，攻击者简单的改变调色板就可以破坏隐秘信息

改进方法1：对调色板排序（例如按照亮度排序），然后对调色板用LSB方法

- 排序方法有很多种，例如，给出一个像素值，找到下一个最接近的像素值，如此形成一个序列

改进方法2:把调色板数量倍增，然后微小的修改备份的调色板。

2.4 声音隐写术

声音中隐藏信息更难，因为人类听觉系统对噪音扰动非常敏感。

1. 但人类听觉系统对声音的相位不太敏感，这个事实大量用于声音压缩。
2. 回声隐藏

回声隐藏

$$f(t) = f(t) + \alpha f(t - \Delta t)$$

- 如果要嵌入0，就取 $\Delta t = k$
- 如果要嵌入1，就取 $\Delta t = k$

2.5 视频隐写术

把图片盲水印应用的视频时，水印和载体的信息比的问题就不重要了，但是时间维度上的失真就难以控制。

有一种基于运动矢量化方法，实现把信息嵌入 MPEG 视频流中。原理还是类似基于图像的 DCT 方法。

几种隐写术比较：

方法	示例	埋入信息量	是否有损	鲁棒性	隐蔽性
隐写在频域上		少	有损	很强，抗各种攻击。	很强
隐写在空域上		中	无损	弱	强
图种	copy /b xx.jpg+xx.rar xx.jpg	理论上无上限	无损	弱	几乎无，像素和文件大小不匹配、扫描内容都可以发现

3. 隐写术算法评价指标

- 不可感知性
- 失真度
- 鲁棒性
- 隐秘性
- 嵌入信息的量。一般来说，嵌入信息越多，鲁棒性越低
- 水印强度，一般与水印的可感知性也是一对 trade off

鲁棒性

嵌入水印后，应当对以下操作鲁棒：

- 信号增强：锐化、增强对比度、色彩校正、伽马校正
- 噪声：高斯、均衡、斑点、蚊状
- 滤波：低通、高通、带通
- 有损压缩：JPEG（图像），MPEG（视频），MPEG（音频）
- 局部或全局仿射变换：平移、旋转、缩放、剪切
- 压缩
- 合成：标志插入、加入布景
- 格式转化：例如GIF转JPEG
- D/A和A/D变换，例如：先打印后扫描，转成模拟信号再转回数字信号
- 多重水印：再次被打上水印，会使一些隐写水印“变淡”甚至被破坏
- 统计平均：例如，攻击者找到同一个图片的加入不同水印的版本，做一下平均后，往往可以使所有水印都被破坏
- 马赛克

失真度的度量

正常的图像来说，颜色会逐渐变化，但很少会出现“1比特变化”这种情况。

- 打上水印的图和原图之间的Lp范数 $L_p = \left(\frac{1}{N} \sum |I - I'|^p \right)^{\frac{1}{p}}$
- 信噪比: $SNR = \frac{I}{I'}$ • p=1, p=2 是最常用的
- 参考以上两条的思路, 可以定义很多指标
- 直方图相似性。先算出两个图的直方图, 然后求出相似性。
- 人工方法: 就是找十来个人打分, 针对能不能察觉到水印做打分

4. 攻击隐写术

攻击方法的分类

- 被动攻击: 接受载体, 分析载体
 - 隐写分析: 分析载体是否被隐写
 - 破解隐写术: 确定载体被隐写后, 提取秘文
- 主动攻击: 截取载体, 处理载体 (例如压缩图片, 要保证人感知不到变化), 使密文被去除 (或者没去干净但无法再解出来), 继续传输载体。

隐写分析

隐写分析的目的是分析载体是否被隐写。

- 属于模式识别或机器学习分类问题。
- 用机器学习技术, 察觉特性很小的偏差
- 隐写算法在嵌入秘密数据时, 必定修改原数据。载体的某些统计特征必然变化。
- 通用隐写分析针对多种隐写术。专用隐写分析针对某个隐写术。

估计秘密信息长度、破译秘密信息, 属于量化分析和解密范畴。

隐写分析方法

- 视觉隐写分析。(Visual steganalysis)
- 结构隐写分析 (Structural steganalysis)
- 统计隐写分析 (Statistical steganalysis): 定义一套随机变量, 做参数统计建模
- 学习隐写分析 (Learning steganalysis): 用大量的隐写后的文件、未隐写的文件做为样本, 做一个二分类机器学习模型。

视觉隐写分析 的例子:

某个隐写术利用了人类视觉的不可感知性。对某个图做二值化, 发现有大量噪点, 判断用了隐写术, 如下图:



(左边是原图二值化，右边是打上隐写的图二值化，显然右边隐写了信息)

结构隐写分析：

- 任何隐写工具都会引入某种特征。
- 早期某种文本隐写术，利用html的特点
- 文本，调整文本颜色、大小、位置
- 链接：不用下划线，且鼠标指针经过链接时不变化
- 注释：html注释

主动攻击

- 降低图像质量
- 再次加入水印，往往可以削弱原有水印，甚至使之失效。一些版权保护的商业软件为了应对这种攻击，会拒绝对已有水印的图再次加入水印
- 取平均。例如，同一个视频加入不同的水印，如果攻击者获得了多个加水印后的版本，只需要把它们做个相加，往往就抹除隐水印。
- 减法攻击。甲的原图是 I ，水印是 x ，嵌入图是 $I+x$ 。攻击者乙的水印是 y ，然后声称 $I+x-y$ 是自己的原图。

参考资料

冈萨雷斯：《数字图像处理》，电子工业出版社

S.Katzenbeisser：《信息隐藏技术：隐写术与数字水印》，人民邮电出版社

刘粉林：《数字图像隐写分析》，机械工业出版社出版