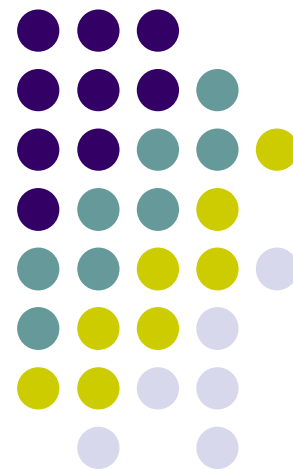




信息隐藏技术基础

王莘



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

上节重点

1. 隐写分析的目的是什么？
2. 使用卡方检验进行隐写分析的原理是什么？
3. 什么是正向翻转，负向翻转和0翻转？
4. 描述RS隐写分析过程，阐述其原理
5. 怎样判断DCT系数直方图是否改变？



摘要

- 抗隐写分析的LSB的隐写
 - 直方图补偿法
 - 随机翻转嵌入法
- 抗隐写分析的JPEG图像隐写



摘要

- 抗隐写分析的LSB的隐写
 - 直方图补偿法
 - 随机翻转嵌入法
- 抗隐写分析的JPEG图像隐写



针对LSB的隐写分析-- χ^2

χ^2 分析方法的特点

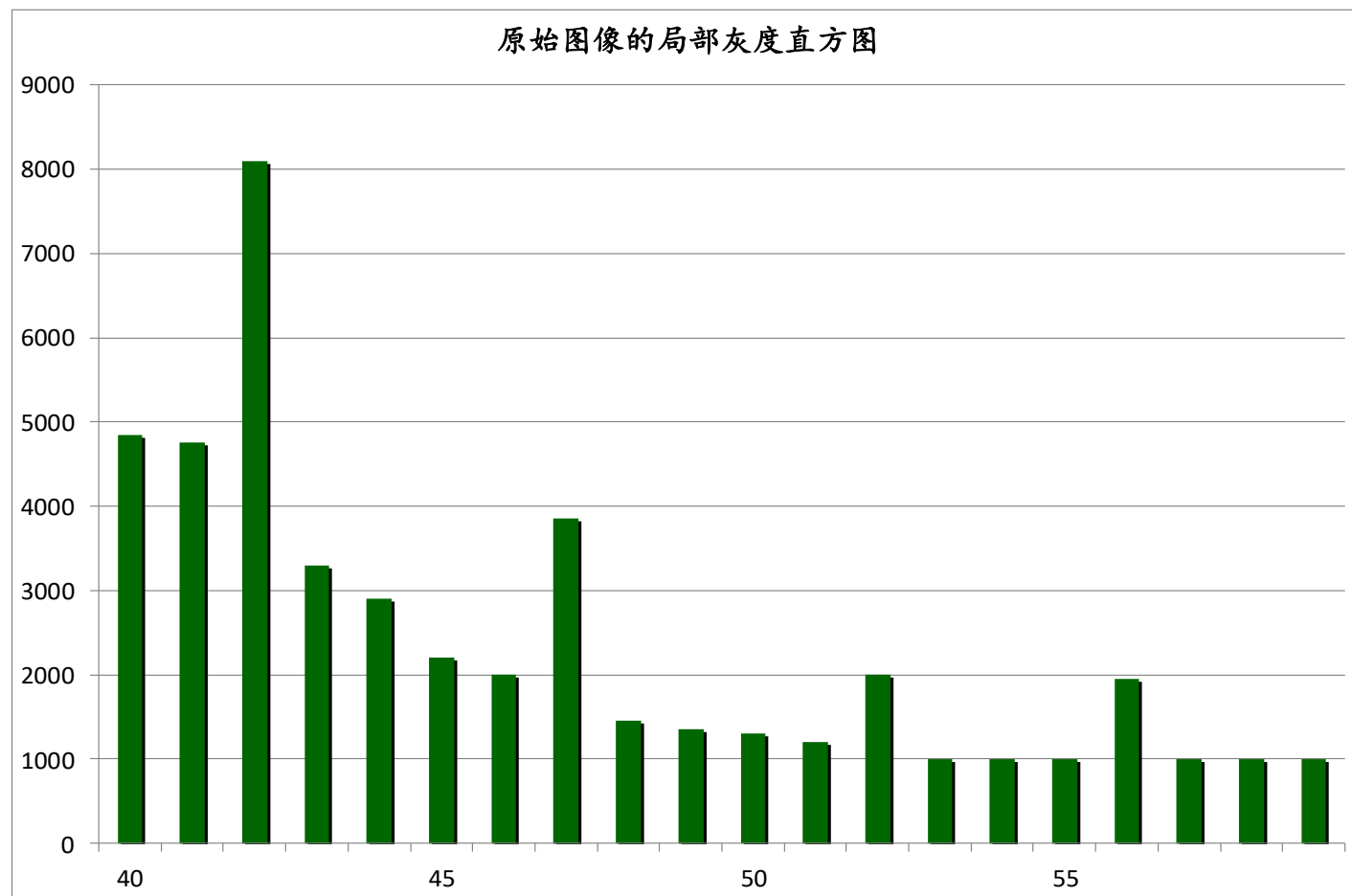
- 原理简单清晰
- 易于实现, 检测效果较好
- 只能针对特定的LSB嵌入策略
- 只能针对高嵌入率的情况
- 对部分伪随机嵌入无效



针对LSB的隐写分析-- χ^2

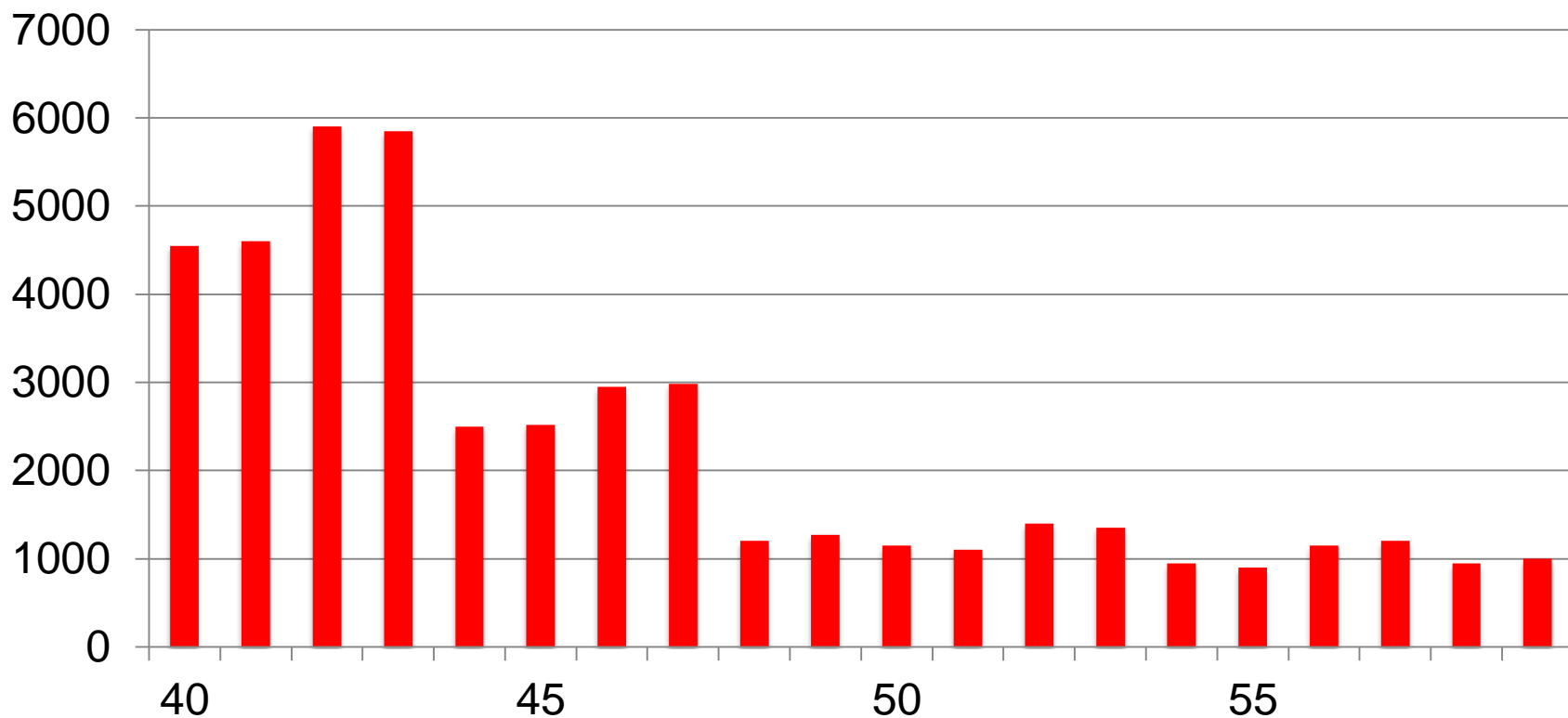


针对LSB的隐写分析-- χ^2



针对LSB的隐写分析-- χ^2

隐写图像的局部灰度直方图

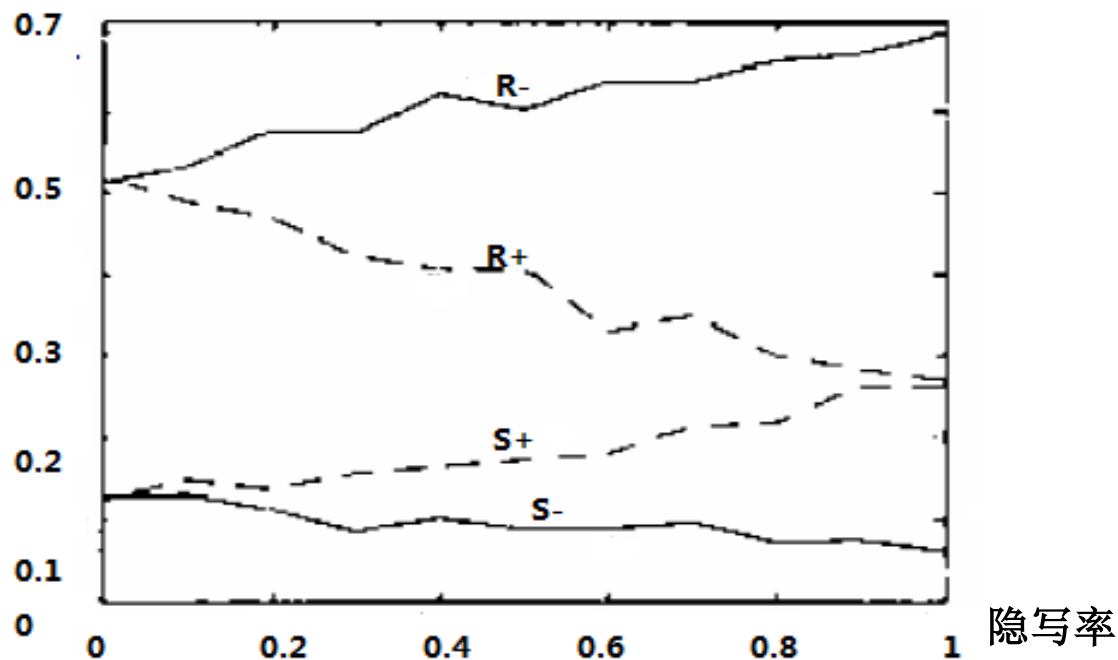


针对LSB的隐写分析--RS

如果待检测图像没有经过LSB密写，那么无论是经过非正翻转还是非负翻转，会等同的增加图像块的混乱度，都应该满足以下规律： $R_m \approx R_{-m}$ ， $S_m \approx S_{-m}$ ，且 $R_m > S_m$ ， $R_{-m} > S_{-m}$ 。



针对LSB的隐写分析--RS



不同隐写率下的R, S



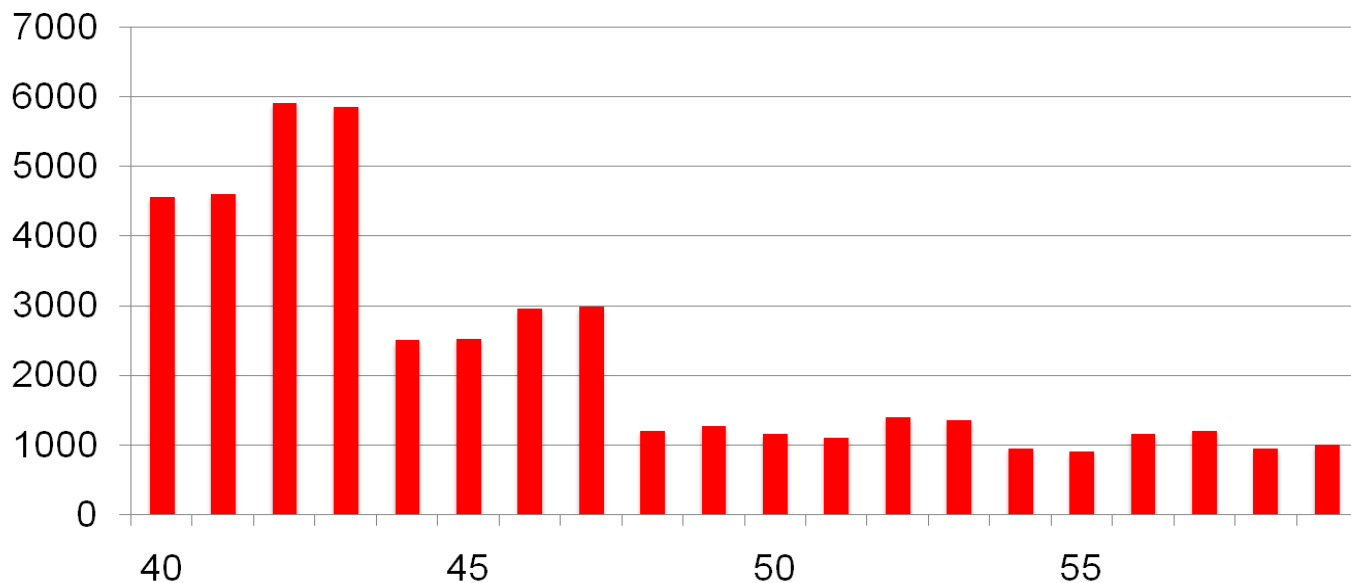
直方图补偿隐写方法



直方图补偿隐写方法

通过在隐写后的图像中进行额外的操作，将隐写引起的直方图失真失真补偿回来。主要补偿方式是通过调整未嵌入秘密信息的像素值。

隐写图像的局部灰度直方图



直方图补偿隐写方法

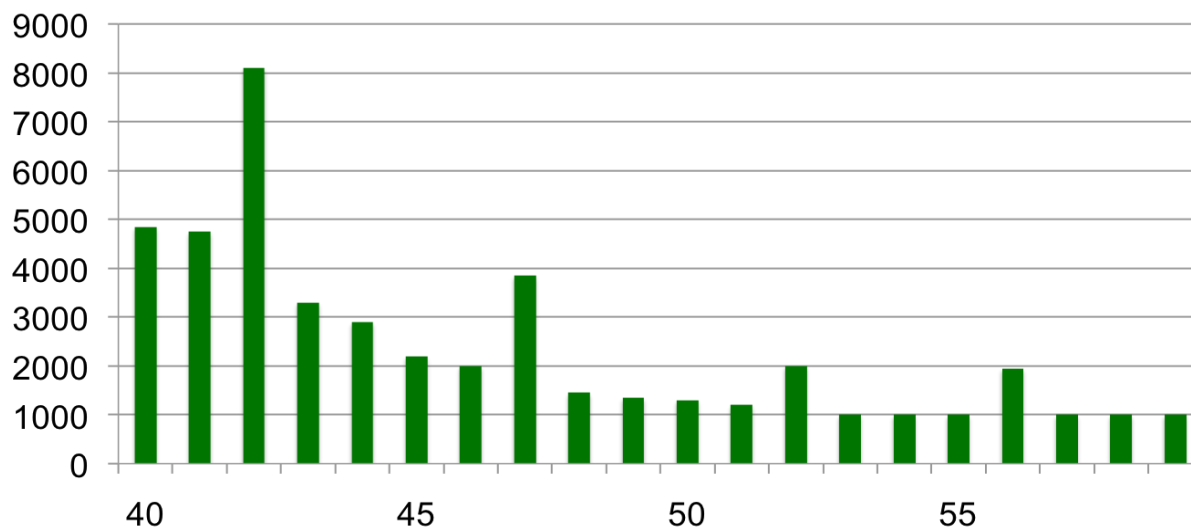
$$(1-\alpha)f_{2i+1} \geq \frac{\alpha}{2}(f_{2i} - f_{2i+1})$$

$f_i \dots$ 原始图像像素数目

$\alpha \dots$ 秘密信息嵌入率

$$\Rightarrow \alpha \leq \frac{2f_{2i+1}}{f_{2i} + f_{2i+1}}$$

原始图像的局部灰度直方图



直方图补偿隐写方法

1. 统计原始图像的灰度直方图，主要确定 f_{2i} 和 f_{2i+1} 的数量。
2. 计算秘密信息隐藏率 α
3. 将所有的像素值排成一个向量。向量长度 l 。
4. 秘密信息嵌入到 $\lfloor \alpha \times l \rfloor$ 为止。
5. 如果 f_{2i} 变大，将 $\lfloor \alpha \times l \rfloor + 1$ 之后的像素中的值为 $2i$ 的像素变为 $2i+1$ ，尽可能保持直方图不发生变化；反之类似。



摘要

- 抗隐写分析的LSB的隐写
 - 直方图补偿法
 - 随机翻转嵌入法
- 抗隐写分析的JPEG图像隐写



抗RS分析的隐写方法

——随机翻转嵌入法



抗RS分析的隐写方法

对LSB方法进行修改，在嵌入数据时不但可将 $2i$ 改为 $2i+1$ ，或者将 $2i+1$ 改为 $2i$ ，也可以将 $2i$ 改为 $2i-1$ ，或者将 $2i-1$ 改为 $2i$ 。



抗RS分析的隐写方法

设秘密信息为 w ，对应隐藏该位的像素灰度值为 $x(i, j)$ 。

如果 w 与 $x(i, j)$ 的最后一位相同，那么不改变原始数据。

当 w 与 $x(i, j)$ 的最后一位不同，对 $x(i, j)$ 进行调整。

$$x(i, j) = \begin{cases} x(i, j) - 1, & T \leq 0, \quad 0 < x(i, j) < 255 \\ x(i, j) + 1, & T > 0, \quad 0 < x(i, j) < 255 \\ x(i, j) - 1, & x(i, j) = 255 \\ x(i, j) + 1, & x(i, j) = 0 \end{cases}$$

$$T = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} x(u, v) - 9x(i, j)$$



抗RS分析的隐写方法

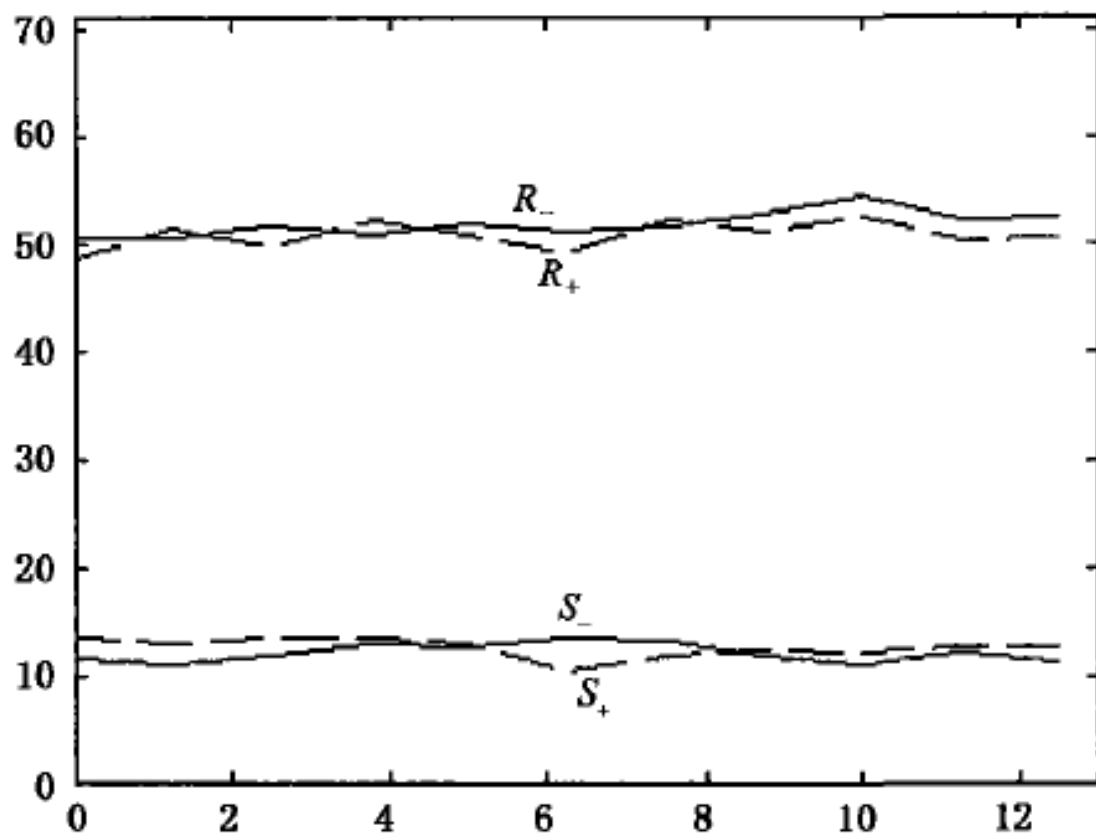
改进的方案与原有的LSB隐写技术相比，修改的幅度并没有增大，所以图像的失真度还是相同的。

会有大约一半像素的最低比特位与秘密信息相同而不发生变化，其余的发生变化。发生变化的像素灰度值中，又会有大约一半的像素值有 $2i$ 变为 $2i+1$ 或由 $2i+1$ 变为 $2i$ 。而另一半的像素值有 $2i$ 变为 $2i-1$ 或由 $2i-1$ 变为 $2i$ 。使用RS统计分析时，无论使用 F_1 翻转还是 F_{-1} 翻转都会以同样程度增加图像混乱度。

对秘密信息的提取非常简单，只要将秘密信息的灰度最低位取出即可。



抗RS分析的隐写方法



隐写率



摘要

- 抗隐写分析的LSB的隐写
 - 直方图补偿法
 - 随机翻转嵌入法
- 抗隐写分析的JPEG图像隐写



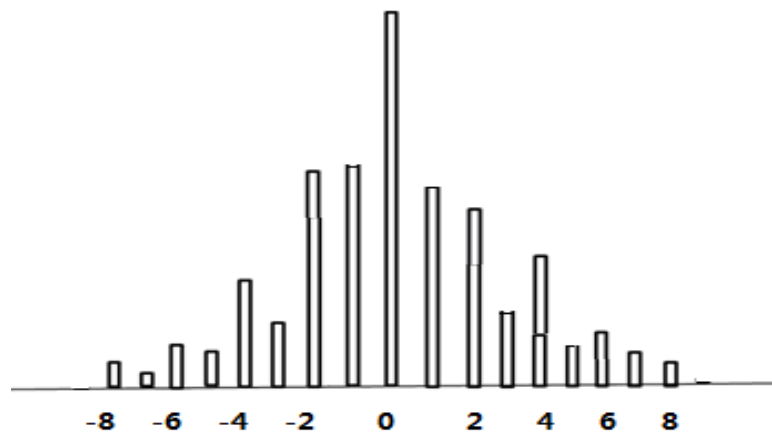
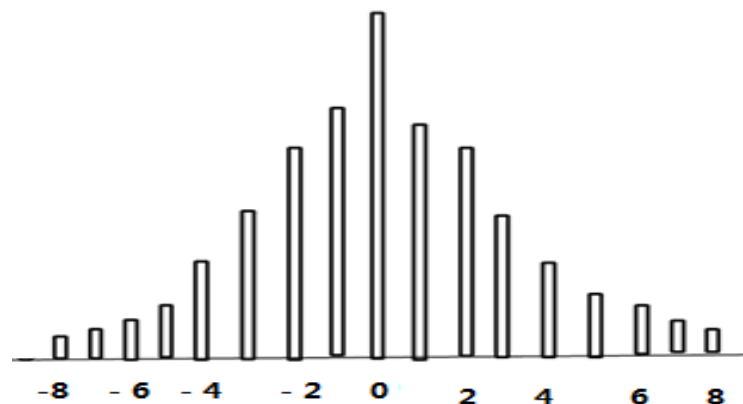
F5隐写方法



F3隐写分析

如果秘密信息与DCT的LSB相同，
便不作改动；如果不同，将DCT系
数的绝对值减小1，符号不变。

当原始值为+1或-1且预嵌入秘密信
息为0时，将这个位置归0并视为无
效，在下一个DCT系数上重新嵌入。



F5隐写算法

- F5隐写是在JPEG隐写的基础上添加混洗技术与矩阵编码技术而成，它进一步提高了密写技术的性能。混洗的目的是为了使秘密信息分布在整幅图像中，矩阵编码则是为了提高嵌入效率。
- F5的主要特点是在 2^k-1 个载体数据中嵌入 k bit的秘密信息，而最多只需要修改一个载体数据。



混洗技术

考虑非0值DCT系数的个数大于秘密信息长度时的情况。如果将秘密信息顺次地嵌入在DCT系数上，那么秘密信息将会集中于图像的某一部分，可能导致图像质量分布不均衡。为了避免这种情况的发生，可以采取混洗的方法，即将DCT系数打乱原有顺序重新排列，然后逐个处理嵌入秘密信息。嵌入后再恢复原来的次序，得到含秘图像。



矩阵编码

无论待嵌入的 k bit秘密信息与载体数据有多么不匹配（ 2^k-1 种不匹配的可能），矩阵编码总能从 2^k-1 个载体数据中找到一个并将其修改，使得所有 k bit秘密信息与载体数据完全匹配。



矩阵编码

k=3时的矩阵编码如下图所示：

	x2	x1	x0
a0	0	0	0
a1	0	0	1
a2	0	1	0
a3	0	1	1
a4	1	0	0
a5	1	0	1
a6	1	1	0
a7	1	1	1
	xor		
	c2	c1	c0

第一行x2、x1、x0表示三个待嵌入的秘密比特。第一列a0-a7是8个载体数据(a0为了好理解放在这里，实际上并没有用到，a0-a7使用正奇数负偶数规则)。中部灰色区域的每一行0、1数据是左边首行载体数据下标的二进制表示。下面最后一行的c2、c1、c0是矩阵编码的结果，它们三个二进制数所对应的十进制整数c就是要修改的载体数据的下标。



矩阵编码

	x2	x1	x0
a0	0	0	0
a1	0	0	1
a2	0	1	0
a3	0	1	1
a4	1	0	0
a5	1	0	1
a6	1	1	0
a7	1	1	1
	xor		
	c2	c1	c0

首先左边第一列载体数据依次乘到灰色的每一列中。然后灰色的每一列，都各自通过异或“累加”起来，结果分别为c2、c1、c0。



矩阵编码

	x2	x1	x0
a0	0	0	0
a1	0	0	1
a2	0	1	0
a3	0	1	1
a4	1	0	0
a5	1	0	1
a6	1	1	0
a7	1	1	1
	xor		
	c2	c1	c0

横着看每一行的运算，可以发现矩阵编码的本质就是让 2^k 个载体数据分别以所有 2^k 种可能的方式去参与和影响 k bit 秘密信息的嵌入。只需要修改这个载体数据，那么这种匹配或不匹配情况就会被消除，成为完全匹配的。



矩阵编码

x_1, x_2 : 待嵌入的秘密信息

a_1, a_2, a_3 : 载体数据

如果:

$$x_1 = a_1 \oplus a_3$$

$$x_2 = a_2 \oplus a_3$$

不改变原始数据

$$x_1 \neq a_1 \oplus a_3$$

$$x_2 = a_2 \oplus a_3$$

改变 a_1 的LSB

$$x_1 = a_1 \oplus a_3$$

$$x_2 \neq a_2 \oplus a_3$$

改变 a_2 的LSB

$$x_1 \neq a_1 \oplus a_3$$

$$x_2 \neq a_2 \oplus a_3$$

改变 a_3 的LSB



F5隐写实现方案

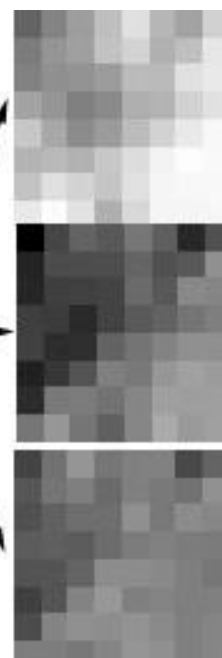
- 解码JPEG图片，获得量化后的DCT系数。
- 混洗DCT系数，混洗的方法作为密钥。
- 确定 k ，并计算 $n=2^k-1$ 。



F5隐写分析



F5隐写分析


$$\begin{bmatrix} 81 & 116 & \dots & 133 & 170 \\ 104 & 130 & \dots & 159 & 201 \\ \vdots & \vdots & \backslash & \vdots & \vdots \\ 155 & 189 & \dots & 218 & 218 \\ 197 & 221 & \dots & 216 & 217 \end{bmatrix}$$
$$\begin{bmatrix} 12 & 74 & \dots & 51 & 95 \\ 37 & 83 & \dots & 99 & 155 \\ \vdots & \vdots & \backslash & \vdots & \vdots \\ 60 & 133 & \dots & 167 & 168 \\ 132 & 163 & \dots & 164 & 169 \end{bmatrix}$$
$$\begin{bmatrix} 75 & 134 & \dots & 90 & 124 \\ 104 & 137 & \dots & 136 & 165 \\ \vdots & \vdots & \backslash & \vdots & \vdots \\ 96 & 155 & \dots & 150 & 156 \\ 149 & 149 & \dots & 150 & 156 \end{bmatrix}$$


F5隐写分析

分块特性分析

- 由于JPEG分块量化并实施F5隐写，这会导致块边界的不连续性增加。这两中现象会受到k值的影响，k值越小，这两种现象表现得越明显。
- 计算分块效应值得公式为：

$$B = \sum_{i=1}^{\lfloor (M-1)/8 \rfloor} \sum_{j=1}^N |g_{8i,j} - g_{8i+1,j}| + \sum_{i=1}^M \sum_{j=1}^{\lfloor (N-1)/8 \rfloor} |g_{i,8j} - g_{i,8j+1}|$$



F5隐写分析

- 也就是以8x8分块为一个网格，所有横网格线两边的像素的差值之和，加上所有竖网格线两侧的像素的差值之和，其中M和N是图像的行数和列数， g_{ij} 表示图像第i行j列的数据。
- 通过计算并比较参考图像和待测图像的B值，就可以大致判断参考图像是否被修改过。参考图像的B值接近原始图像，待测图像如果经过了隐写，那么B值会比参考图像显著增大。



谢谢！



本节重点

1. 直方图补偿法原理
2. 直方图补偿法中怎样计算秘密信息嵌入率
3. 随机翻转嵌入法的原理
4. 随机翻转嵌入法怎样处理边界溢出问题
5. F5隐写算法使用了哪两种重要方法，其目的是什么？
6. F5隐写后图像的哪种统计特性发生了变化？
7. 使用F5隐写算法进行秘密信息嵌入，计算嵌入信息后的像素值

x_1, x_2 : 0 1

x_1, x_2, x_3 : 1 0 1

a_1, a_2, a_3 : 123 124 125

a_1, \dots, a_7 : 13 17 16 16 19 21 20

