基于 SPIHT 的图像加密与压缩关联算法*

杨华千[1)2)3)†] 廖晓峰[1)] Kwok-Wo Wong[2)] 张伟[3)] 韦鹏程[3)]

1) (重庆大学计算机学院, 重庆 400044)

2) (Dept of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong)

3) (重庆教育学院网络管理中心, 重庆 400067)

(2011 年 3 月 14 日收到; 2011 年 6 月 20 日收到修改稿)

为了研究图像压缩与加密同步进行问题, 本文提出了一种在变换域下的图像加密与压缩关联算法在该算法中, 加密过程发生在小波变换与 SPIHT 编码之间它充分利用了离散小波变换和基于层次树的集合划分 (set partitioning in hierarchical trees, SPIHT) 编码属性, 扩散过程被限制在单个子带内部. 此外, 混淆过程保留了 SPIHT 编码中两个最重要的位和符号位, 它包含了图像的重要信息. 实验结果表明, 算法具有良好的安全性、图像重构视觉质量以及很高的加/解密速度.

关键词: 密码系统, 压缩, 离散小波变换 (DWT), 基于层次树的集合划分 (SPIHT)

PACS: 05.45.Gg

1 引 言

近年, 随着因特网的快速发展, 网络上传输图像的数量越来越多、尺寸越来越大, 这对网络带宽带来了很大的压力. 另一方面, 出于安全原因需要对图像进行加密保护图像的隐私. 为了满足互联网上高速安全地传输图像, 传统的做法是将图像的压缩与加密分成两步来完成. 在这个过程中, 一般是先对图像进行压缩然后加密, 反过来则不会取得很好的压缩效果. 然而, 这样做将失去压缩与加密同步完成所具有的设计灵活与计算简化的一些优势. 一个更好的办法是将压缩与加密关联起来同步完成, 这正成为当前的一个研究热点 [1−5].

目前, 许多研究文章已经揭示了混沌系统与密码系统之间的紧密关系 [6−9]. 此外, 离散小波变换也已经广泛应用于图像压缩中 [10,11]. 因此, 组合 DWT 的压缩特性与混沌系统的密码特性是一个有前途的研究领域. 文献 [12] 提出了一种采用了混杂 HARR 小波编码和混沌掩码的对称加密算法. 然而, 该算法仅仅采用了 HARR 子小波基 $\{h_0, h_1, h_2, \cdots, h_{n-1}\}$ 的组合形式来搅乱明文位流的位置而并没有改变明文位流的值. 在文献 [13] 中, 作者提出了一种新的私钥密码系统, 其加密与解密作用在非线性有限域小波变换的分析子带和综合子带上, 但是作者并没有研究这种算法的压缩性能. 文献 [14] 研究了一种感知加密算法, 它作用在基于 SPIHT 压缩编码的基础上. 在该算法中, 来自于四叉树中同一父结点的四个小波系数位置被加密算法搅乱了, 在加密过程中其系数值并没有变化. 然而, 根据 Shannon 理论 [15], 一个好的密码系统应该包含两个过程: 扩散与混淆, 否则, 该密码系统存在潜在的危险, 容易受到选择明文攻击 [7].

在文献 [16] 里, Lin 等提出了一个混沌图像编码算法来加密 SPIHT 编码器产生的编码位流. 该文提出的算法本质上是一种流密码算法, 它并没有充分利用 SPIHT 的编码特性. SPIHT 编码算法具有理想的渐进式传输性能, 同时还具有良好的压缩比. 在 SPIHT 中, 由于最重要位 (most significant bits, MSBs) 包含了重要的图像信息, 它们应该出现在压缩流的开始部分并首先传到解码器. 因此, SPIHT 编码算法需要先对系数进行排序, 并且首先传输最重要位以便实现渐进传输与压缩.

本文提出了一种图像加密与压缩关联算法. 在

* 香港特别行政区研究拨款委员会资助 (批准号: CityU 122308), 重庆市科委自然科学基金 (批准号: CSTC, 2010BB2279), 重庆市教委科学技术研究项目 (批准号: kj091501, kj091502, kj101501) 和中国博士后基金 (批准号: 2011M501391) 资助的课题.

† E-mail: maitostorm@163.com

该算法中, 加密过程被安排在小波变换与 SPIHT 压缩编码之间. 加密过程中, 混淆过程被限制在单个小波子带里, 该子带包含了图像的某部分细节信息. 同时, 最重要位和符号位保持不变.

2 SPIHT 编码算法概述

SPIHT 算法能够生成一个嵌入位流, 使接收的位流在任意点中断时, 都可解压和重构图像, 具有良好的渐进传输特性在系数子集的分割和重要信息的传输方面采用了独特的方法, 利用了小波变换系数的量级有序、集合划分、有序的位平面传输以及图像小波变换下的自相似原理. 能够在实现幅值大的系数优先传输的同时, 隐式地传送系数的排序信息. 其主要步骤如下 [10,17]:

1) 采用适当的小波滤波器对图像进行小波分解, 然后用固定比特位数表示变换后的系数 $c_{i,j}$, 最高位是符号位, 其余位是数量位, 最不重要位排在最下面.

2) 排序扫描: 传输满足条件 $2^n \leqslant |c_{i,j}| < 2^{n+1}$ 的系数 $c_{i,j}$ 的个数 l, 及对应系数的 l 对座标和 l 个符号位.

3) 精细扫描: 传输满足 $|c_{i,j}| \geqslant 2^{n+1}$ 的所有系数的第 n 个最重要比特位. 它们是第 2 步排序过程中选出来的系数.

4) 迭代: 如需继续迭代, 则 n 减少 1, 转步骤 2).

排序是主要步骤, 其主要任务是选择满足 $2^n \leqslant |c_{i,j}| < 2^{n+1}$ 的系数. 对于给定的 n 值, 如果系数 $c_{i,j}$ 满足 $|c_{i,j}| \geqslant 2^n$, 则认为该系数是重要的, 否则认为是不重要的. 在第一次迭代过程中, 只有少数系数是重要的. 然而, 由于 n 一直在减小, 随着迭代的深入, 系数将越来越多. 排序过程必须确定哪些重要系数满足 $|c_{i,j}| < 2^{n+1}$ 并且传递这个坐标给解码器. 这是 SPIHT 算法最重要的一部分.

3 图像加密与压缩关联算法

在本文提出的算法中, $x_1, x_2, \cdots, x_K, x_{K+1}$ 是以双精度浮点格式表示的 Logistic 映射的秘密初始值. K 是 DWT 分解的级数. 这些值被用来产生扩散过程的初始值和混淆过程的二进制流.

3.1 基本结构

本文提出的关联加密与压缩算法的算法框图如图 1 所示. 加密过程发生在 DWT 分解之后 SPIHT 压缩之前. 整个加密过程由两个阶段组成: 第一阶段利用标准映射来实现扩散过程, 在这个阶段, 子带图里的所有小波系数作为一个整体被扰动, 扰动过程中其值没有发生改变. 第二个阶段是用 Logistic 映射来实现混淆过程, 在这个阶段满足某些条件的系数将被修改, 继而, 系数的微小改变将扩散到尽可能多的系数上. 最后, 一个附加的掩码操作被作用在所有的加密和压缩比特位上.

3.2 密钥生成与编排

图像的离散小波变换可以使用任何的小波滤波器, 并且能够以任何形式分解图像. 唯一的限制是, 子带里必须要有足够的数据点来覆盖所有的滤波条带.

金字塔分解被证明是一种非常有效的小波分解方法. 它产生水平、垂直和对角三个子带的图像细节数据. 每一级的三个子带都包含了特定尺度的水平、垂直和对角图像特征信息. 因此, 本文使用了这种分解方式. 如图 1 所示, 每一级的每个子带将独立地进行扩散与混淆操作进行加密. 密钥编排如下:

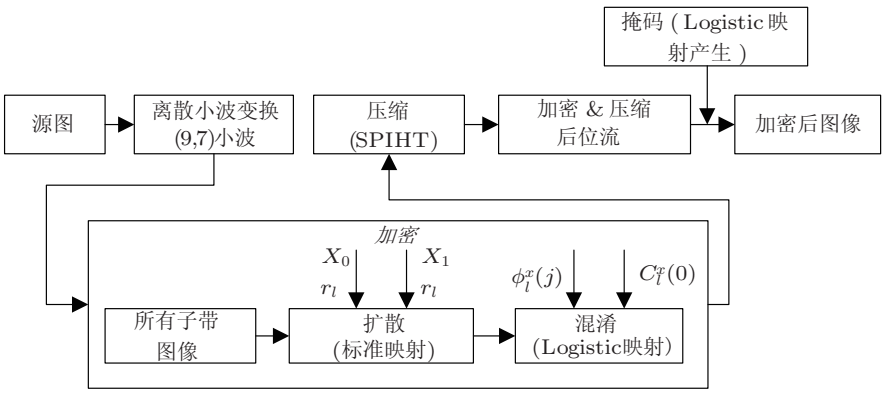

图 1 图像加密与压缩关联算法框图

1) 用 SHA1 算法生成原始图像和初始值 x_1, $x_2,\cdots,x_K$, x_{K+1} 的 160 bit HASH 值 s_0, 如图 2 所示.

(2) 用 SHA1 算法产生 s_0 和 x_l 的 160 bit Hash 值 s_l $(1\leqslant l\leqslant K)$.

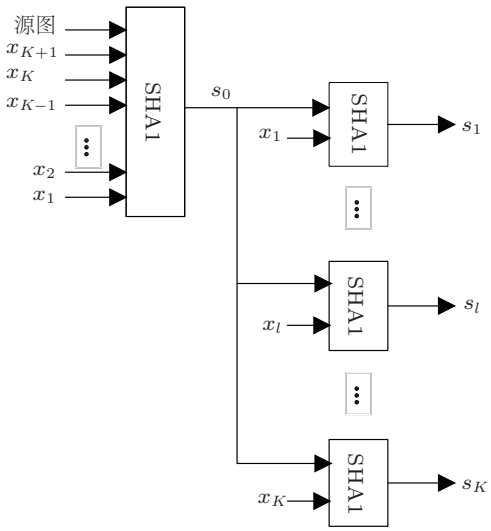

图 2 160 bit HASH 值的生成

在扩散阶段, 标准映射并不会扰动图像左上角的系数, 这可能会成为密码分析中的一个漏洞. 因此, 一个随机数对 $(r_l^{X_0}, r_l^{X_1})$ 被引入进来. 这样, 修改后的标准映射如下:

$$\begin{aligned} s_{k+1} &= \left(s_k + t_k + r_l^{X_0} + r_l^{X_1}\right) \bmod \frac{N}{2^l},\\ t_{k+1} &= \left(t_k + r_l^{X_1} + K_l^c \sin\frac{2\pi\cdot s_{k+1}}{N/2^l}\right) \bmod \frac{N}{2^l}, \end{aligned} \tag{1}$$

此处, (s_k, t_k) 和 (s_{k+1},t_{k+1}) 分别是 $N/2^l\times N/2^l$ 系数矩阵中扩散前和后的位置. 标准映射参数 K_l^c 是一个正整数 $(1\leqslant l\leqslant K)$.

对于第 $l(1\leqslant l\leqslant K)$ 级的三个子带图 LH_l, HH_l 和 HL_l: $K_l^c\leftarrow s_l^{97}\cdots s_l^{108}$. s_l 的其他 bit 按如下方式安排

1) LH_l: $r_l^{X_0}\leftarrow s_l^1\cdots s_l^8$, $r_l^{X_1}\leftarrow s_l^9\cdots s_l^{16}$, $C_l^X(0)=s_l^{65}\cdots s_l^{72}$;

2) HH_l: $r_l^{X_0}\leftarrow s_l^{17}\cdots s_l^{24}$, $r_l^{X_1}\leftarrow s_l^{25}\cdots s_l^{32}$, $C_l^X(0)=s_l^{73}\cdots s_l^{80}$;

3) HL_l: $r_l^{X_0}\leftarrow s_l^{33}\cdots s_l^{40}$, $r_l^{X_1}\leftarrow s_l^{41}\cdots s_l^{48}$, $C_l^X(0)=s_l^{81}\cdots s_l^{88}$.

当 $l=K$ 时, 对于左上角的子带图

$$\begin{aligned} \mathrm{LL}_K: r_K^{X_0}\leftarrow s_K^{49}\cdots s_K^{56},\quad r_K^{X_1}\leftarrow s_K^{57}\cdots s_K^{64},\\ C_l^X(0)=s_l^{89}\cdots s_l^{96}. \end{aligned}$$

3.3 加密与压缩

本文提出的关联加密与压缩算法步骤如下:

1) 对于指定的明图, 用合适的小波滤波器进行离散小波变换, 得到变换系数.

2) 加密小波变换后每一级包含的三个子带系数和最后一级的左上角子图系数. 加密过程包含如下的两步:

扩散过程: 用 3.2 节中的 (1) 式和参数搅乱每个子图中的所有小波变换系数.

混淆过程: 改变那些绝对值大于等于阈值 $\delta(\delta\geqslant 1)$ 的被扰动过的系数. 其具体操作步骤如下:

如果

$$\lfloor|c_l^X(j)|\rfloor < 2^{\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor} + 2^{(\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor-1)},$$

则用

$$\begin{aligned} C_l^X(j) =&(\lfloor|c_l^X(j)|\rfloor \oplus \lfloor|C_l^X(j-1)|\rfloor \oplus \phi_l^X(j))\\ &\times \bmod 2^{(\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor-1)}\\ &+ 2^{\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor} \end{aligned} \tag{2}$$

来修改当前被扰动的系数, 否则用

$$\begin{aligned} C_l^X(j) =&(\lfloor|c_l^X(j)|\rfloor \oplus \lfloor|C_l^X(J-1)|\rfloor \oplus \phi_l^X(j))\\ &\times \bmod 2^{(\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor-1)}\\ &+ 2^{\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor}\\ &+ 2^{(\lfloor\log_2\lfloor|c_l^X(j)|\rfloor\rfloor-1)} \end{aligned} \tag{3}$$

来修改系数值.

然后, 所有的系数 $C_l^X(\cdot)$ 按下式再做一次处理:

$$C_l^X(\cdot)=\begin{cases} -\left(C_l^X(\cdot)+|c_l^X(\cdot)|-\left\lfloor|c_l^X(\cdot)|\right\rfloor\right), \\ \qquad c_l^X(\cdot)<0, \\ C_l^X(\cdot)+|c_l^X(\cdot)|-\left\lfloor|c_l^X(\cdot)|\right\rfloor, \\ \qquad c_l^X(\cdot)>0. \end{cases} \tag{4}$$

此处, $C_l^X(j)$ 是系数 $c_l^X(j)$ 的加密值. $C_l^X(j-1)$ 是前一个加密值. $C_l^X(0)$ 是如第三节 B 所示的秘密初始值. 函数 $\lfloor\cdot\rfloor$ 实现向下取整, 即返回比输入值小的最大整数. 位抽取函数 $\phi_l^X(\cdot)$ 实现从 Logistic 映射返回值中抽取一定数量的 bit 位.

$$\tau_{j+1}(x)=\mu\tau_j(x)(1-\tau_j(x)),\quad x\in I=[0,1], \tag{5}$$

此处, $\tau_j(x)$ 是 Logistic 映射的第 j 次迭代返回值, 其初始值为 x_l, 值 x 按如下方式表示:

$$x=0.\quad b_1b_2\cdots b_i\cdots,\ x\in[0,1],\ b_i\in\{0,1\}.$$

本文抽取的是小数点后的第 $b_4b_5\cdots b_{15}$ 位, 共 12 bit.

3) 用第 2 节描述的 SPIHT 编码算法压缩加密后的小波系数. 输出的二进制序列用 m 表示, m 中除了包含图像数据信息外, 还包含了 3 个字节的诸如图像大小、小波变换级数、压缩比等头部信息.

4) 掩掉加密并压缩的位流 m. 类似步骤 2 所示, 用 (5) 式定义的 Logistic 映射, 一个与位流 m 等长的比特被抽取了出来, 其初始密钥是 x_{K+1}, 从 Logistic 映射的每一次迭代中抽取其中的 $b_8\ b_9\ \cdots b_{15}$ 共 8 个 bit. 掩码过程如下:

$$Z_i = Z_{i-1} \oplus m_i \oplus \varphi_i, \quad i = 1, 2, \cdots . \tag{6}$$

此处, Z_{i-1} 和 Z_i 分别是第 $i-1$ 和 i 个 8 bit 的掩码流. m_i 是序列 m 的第 i 个字节. φ_i 是从 Logistic 映射的第 i 次迭代中抽取的 8 bit 位流. 当 $i = 1$ 时, $Z_{i-1} = Z_0$ 是秘密分配的初始 8 bit 位流. 序列 m 的头部信息没有被掩掉.

4 实验结果

本节我们进行了大量的仿真实验来测试算法的各种性能. 实验过程中, 采用 (9,7) 不可逆浮点小波变换, 小波分解级数 K 为 6. 共需要 7 个 Logistic 映射初始值. 其中 $x_1, x_2, \cdots, x_6$ 被用在每一小波分解级上, x_7 用在最后的掩码过程中 (如 3.3 节第 4 步所示). 随机选择的 7 个初始值如下: $x_1 = 0.394857698348593$, $x_2 = 0.762757068948018$, $x_3 = 0.689837458934875$, $x_4 = 0.464350945435978$, $x_5 = 0.865798324582347$, $x_6 = 0.654587439865794$, $x_7 = 0.424398752348759$. 此外, $Z_0 = (147)_{10} = (10010011)_2$. SPIHT 算法的 Matlab 程序见文献 [18].

4.1 重构图像的视觉质量

实验过程中, 用一些大小为 512 × 512 像素的标准 8 bit 灰度级图像 (如 Lena, Barbara, Boat, Peppers 和 Baboon), 在不同的压缩比 (compression ratio, CR) (CR∈ $\{0.250, 0.375, 0.500, 0.625, 0.750, 0.875, 1.000\}$, 单位为 bpp (bit per pixel), 即表示每个像素所需的比特数) 和阈值 δ ($\delta \in [1, 32, 128, 256, 512]$) 下进行了测试. 测试结果如表 1 和图 3. 实验结果表明, 在相同的压缩比下, 本文提出的加密与压缩关联算法比单纯的 SPIHT 压缩算法有略低的 PSNR, 但在有损压缩下, 这种方法的视觉质量是可接受的. 造成这种现象的原因主要是在扩散过程中, 部分地打乱了 SPIHT 压缩算法中小波系数的空间方向树的父子关系. 同时, 在混淆过程中也部分地搅乱了 SPIHT 的排序和精细扫描过程. 由于扩散被限制在每一个小波子带的内部, 因此, 这种扭曲是不重要的、可接受的. 此外, 在混淆过程中, 保持了最重要的两个位和符号位不变, 但这几个位包含了图像的最重要的信息.

图 3 图像视觉质量比较 (a) 原始 Lena 图; (b) 只进行压缩的重构图像 (CR=0.250 bpp); (c) 重构图像加密与压缩关联 ($\delta = 1$, CR=0.250 bpp); (d) 重构图像加密与压缩关联 ($\delta = 512$, CR=1.000 bpp)

4.2 密钥空间

一个安全的图像加密算法应该有充分大的密钥空间来阻止暴力攻击, 即通常所说的穷举攻击. 本文提出的算法的密钥空间大小估计如下:

在 K 级金字塔小波分解过程中共得到了 $3 \times K + 1$ 个子带图像. 子带图像的宽或高为 $N/2^l$, 其中, l 是所在的级, N 是原始图像的宽或高. 正如 3.2 节所描述, 在每一级里, 需从 160 bit 的 HASH 值中取出其中的 84 bit 作为 $r_l^{X_0}$, $r_l^{X_1}$, $C_l^X(0)$ 和 K_l^c 的初始值. 此外, 对于第 K 级的左上角 LL_K 子图, 还会在第 K 个 HASH 值中另外再抽取其中的 24 bit. 这样从 K 个 160 bit 的 HASH 值中共抽取了 $(84 \times K + 24)$ bit. 对于理想的 HASH 函数来讲, 其 0/1 分布是平衡的. 这样, 扩散阶段的密钥空间是 $(84 \times K + 24)/2 = 42 \times K + 12$ bit. 在混淆阶段, 位抽取函数 $\phi_l^X(\cdot)$ 从初值为 x_l ($1 \leqslant l \leqslant K$) 的 Logistic 函数的每次迭代值中抽取 12 bit. 根据 IEEE 754 浮点数标准, $\phi_l^X(\cdot)$ 的理论范围是 52 bit. 然而, 由于计算机的数字计算精度限制,

实际实现时被限制在一个较小的密钥空间内. 根据 Kohda 和 Tsuneda 的研究结果 [19], 实际密钥空间是 12 bit, 并且 $\phi_l^X(\cdot)$ 序列是独立同分布的. 因此, 在这个阶段共需要 $K \times 12$ bit. 在掩码阶段, φ_i 需要从初值为 x_{K+1} 的 Logistic 函数的每次迭代值中抽取其中的 8 bit. 尽管, 在这个阶段的密钥空间很小, 但正如第 3.2 节所述, 掩码操作是和扩散与混淆操作相关的. 攻击者并不能分离这两个阶段加以分别攻击. 所以, 算法总的密钥空间是 $42 \times K + 12 + 12 \times K + 8 = 54 \times K + 20$ bit. 当 $K = 6$ 时, 密钥空间将达到 344 bit.

4.3 抗线性攻击与差分攻击分析

线性攻击和差分攻击是两种基于强特征表现的经典密码分析方法. 通过组合这些特征建立明文、密文和密钥之间的关系来进行攻击. 因此, 任何一个密码系统应该具有如下的三个重要属性来维持较高的安全性 [7]: 1) 明文与密文之间的映射是随机的, 仅仅依赖于密钥. 2) 密文对明文是高度敏感的. 3) 密文对密钥是高度敏感的. 这意味着密钥或明文的一个 bit 的变化都应该导致完全不同的密文.

1) 密钥敏感性测试

为了测试密钥敏感性, 实验中对密钥做了一微小改变, 即将密钥小数点后的最后 1 位加 1 或减 1, 然后用新的密钥用来加密和压缩同一个图像. 在我们的算法中, 有 7 个初始密钥, 仅仅一个密钥 x_1 从 0.394857698348593 变为 0.394857698348594. 然后, 两个密文序列进行逐位比较, 并计算其 bit 变化的百分比. 结果如图 4 和表 2. 结果表明, 在所有的压缩比和阈值下, 位变化百分比都接近 50%, 这表明密文对密钥是相当敏感的.

表 1 只进行压缩与加密与压缩关联的 PSNR

	只压缩	加密和压缩 (δ)				
CR(bpp)		1	32	128	256	512
			Lena (512×512)			
0.25	33.679	29.554	30.551	31.224	31.379	31.988
0.500	36.824	31.13	32.702	33.45	33.795	34.891
0.75	38.418	31.731	33.943	34.763	35.199	36.829
1.000	39.951	32.022	34.542	35.499	36.019	38.085
			Babala (512×512)			
0.25	27.086	24.519	24.412	25.084	25.393	25.508
0.500	30.847	26.605	26.995	27.831	27.993	28.153
0.75	33.542	28.36	29.326	30.157	30.481	30.752
1.000	35.82	29.147	30.443	31.57	32.026	32.418
			Baboon (512×512)			
0.25	22.804	21.599	21.602	21.773	21.987	22.001
0.500	25.041	23.393	23.464	23.887	23.946	23.959
0.75	27.081	24.497	24.63	25.285	25.367	25.385
1.000	28.550	26.256	26.506	27.016	27.152	27.172
			Pepper (512×512)			
0.25	33.037	28.568	29.649	30.365	30.674	30.104
0.500	35.554	30.446	31.837	32.828	33.385	32.297
0.75	36.678	31.671	33.107	34.297	35.181	33.569
1.000	37.638	31.908	33.554	34.896	35.926	34.069
			Boat (512×512)			
0.25	30.34	27.368	27.946	28.443	28.585	28.684
0.500	33.738	29.138	30.609	31.34	31.673	31.838
0.75	36.349	29.964	31.96	33.024	33.524	33.779
1.000	38.271	30.974	33.484	34.615	35.37	35.772

2) 明文敏感性测试

为了测试明文敏感性, 明文图像不同位置的一个像素被选出来, 然后和 1 进行异或运算. 这个被修改后的图像被同样的密钥加密与压缩. 两个密文序列逐位比较, 计算 bit 变化百分比. 在我们的实验中, 分别选择了左上角、右下角和随机位置 (197, 339) 三个像素. bit 变化平均百分比列在表 3 和图 5 中. 在所有的压缩比和阈值下, 它们的值都接近 50%. 这表明密文对明文是非常敏感的.

表 2 Lena 图 (512×512) 的密钥敏感性测试

CR\δ	1	32	128	256	512
0.25	50.092	50.015	50.246	49.910	49.646
0.375	50.047	49.877	50.323	49.994	49.727
0.500	50.054	49.976	50.258	49.939	49.756
0.625	50.032	49.986	50.159	50.102	49.743
0.75	50.030	50.014	50.112	50.128	49.838
0.875	49.987	50.050	50.101	50.146	49.846
1.000	50.019	50.068	50.072	50.122	49.843

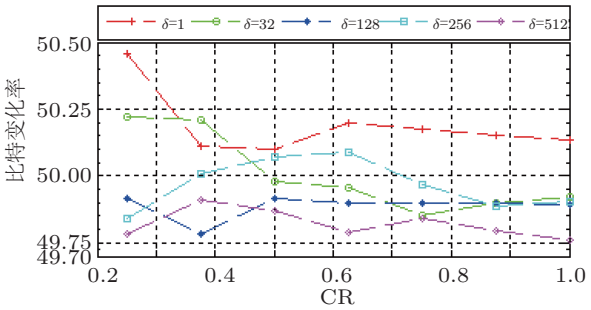

图 4 Babala 图 (512×512) 的密钥敏感性测试

敏感性测试表明, 密文序列对密钥和明图都是非常敏感的, 哪怕是微小的变化. 对密钥和明图的高敏感性能够很好地抵抗差分攻击和线性分析.

4.4 加密与解密速度

在本文提出的算法中, 所有的小波系数将被扰乱一次, 而只有那些大于等于给定阈值 δ 的系数被改变一次. 实验中关于速度性能的比较结果列在表 4 中. 表中给出了算法在不同的阈值 (δ) 和压缩比 (CR) 下加密与压缩关联总时间 (Total, 它包括小波分解时间、加密时间以及压缩编码时间)、加密时间 (Encryption) 以及加密时间占整个算法时间的百分比 (Encry/Total). 实验数据表明:

1) 在给定的阈值下, 随着压缩比 CR 的增大, 总的耗时也逐渐增加而加密时间几乎不变. 这是因为, 当压缩比增大时, 压缩文档中标识每个像素信息所需的 bit 数也增加了, 因此压缩编码的时间随之增加; 而加密时间不变是因为, 加密发生在压缩之前, 加密的小波系数只与阈值 δ 有关.

2) 在给定的压缩比下, 随着阈值 δ 的增大, 总的耗时和加密时间都在减少. 这是因为, 在给定的压缩比之下, 压缩编码的时间几乎不变; 而随着阈值的增大加密的系数个数减少, 因此加密时间减少, 从而也引起总的时间的减少.

表 3 Babala 图 (512 × 512) 的明文敏感性测试

CR\δ	1	32	128	256	512
0.25	50.065	50.160	49.976	49.897	49.772
0.375	50.066	50.084	49.940	49.892	49.860
0.500	50.095	50.101	49.979	49.955	49.951
0.625	50.100	50.020	49.935	50.001	49.939
0.75	50.110	50.006	49.905	50.038	49.978
0.875	50.133	50.012	49.909	50.017	49.964
1.000	50.146	49.986	49.934	50.007	49.978

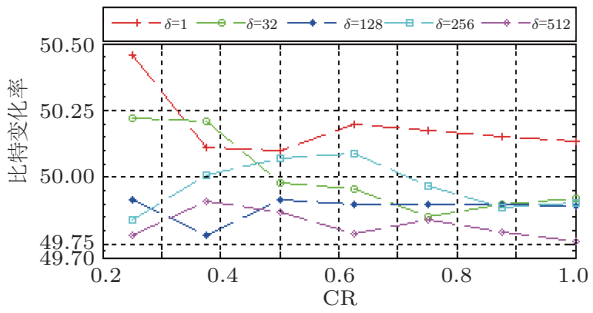

图 5 Lena 图 (512×512) 的明文敏感性测试

3) 加密时间占整个算法时间的相对百分比均小于 50%, 这意味着加密时间不会超过压缩时间.

SPIHT 压缩算法的速度非常快 [10,20], 算法具有潜在的速度优势, 所以这个算法得到了广泛地应用. 注意, 本文使用了 SPIHT 算法提出者 Amir Said, Pearlman 等人在网站 http://www.cipr.rpi.edu/research/SPIHT 上提供的 MATLAB 代码, 全部实验是在 MATLAB 环境下实现的, 并没有进行商业软件的代码优化. 因此, 表 4 中的各项时间略大. 但在对相关代码进行优化后, 对一个 512×512 的 8 bit 灰度图像, 其 SPIHT 压缩时间最大耗时将缩短到 0.64 s[20,Table 1], 结合文本表 4 中的 Encry/Total, 本文的关联算法即使在 CR=1, $\delta = 1$ 时其完成时间也将远低于 1 s($0.64/(1-0.2135) \approx 0.84$ s). 此外, 由于篇幅所限, 本文只列出了 Lena 图的实验结果. 然而, 对于其他图形也有相似的结果.

表 4 加密时间占加密与压缩关联过程总时间的相对百分比 (%)

阈值 δ	时间	压缩比 CR/bpp						
		0.250	0.375	0.500	0.625	0.750	0.875	1.000
1	Total/s	5.657	6.308	8.089	9.879	8.287	10.013	12.294
	Encryption/s	2.593	2.578	2.546	2.875	2.453	2.484	2.625
	Encry/Total/%	45.837	40.867	31.476	29.103	29.601	24.807	21.352
32	Total/s	3.491	4.068	6.175	6.439	7.091	9.152	8.838
	Encryption/s	0.642	0.643	0.655	0.658	0.659	0.659	0.69
	Encry/Total/%	18.392	15.806	10.607	10.219	9.293	7.201	7.807
128	Total/s	2.518	3.095	3.547	4.446	5.407	5.789	7.806
	Encryption/s	0.391	0.39	0.375	0.391	0.406	0.393	0.406
	Encry/Total/%	15.531	12.601	10.572	8.794	7.509	6.789	5.201
256	Total/s	2.250	2.689	3.500	4.424	5.122	5.170	6.473
	Encryption(s)	0.328	0.312	0.343	0.327	0.343	0.326	0.343
	Encry/Total/%	14.579	11.601	9.801	7.392	6.696	6.305	5.299
512	Total/s	2.078	2.648	3.641	4.159	4.614	4.834	6.195
	Encryption/s	0.296	0.294	0.295	0.296	0.312	0.295	0.312
	Encry/Total/%	14.244	11.103	8.103	7.117	6.762	6.103	5.036

表 5 本文算法与分离算法比较

CR	PSNR/dB			加密时间占总的时间比/%		
	本文方法 ($\delta = 512$)	JPEG+AES	JPEG+[20]	本文方法 ($\delta = 512$)	AES/(JPEG+AES)	[20]/(JPEG+[20])
0.250	31.988	29.059	29.059	14.244	13.017	22.011
0.500	34.891	31.276	31.276	8.103	8.207	18.094
1.000	38.085	38.062	38.062	5.036	10.237	20.374

4.5 与传统的压缩加密算法比较

为了与传统的压缩与加密分离算法比较. 本文采用标准的 512×512 像素 8 bit 灰度图像 (Lena 图), 给出了本文提出的关联算法和传统的压缩 (JPEG) 与加密 (AES) 分离算法 (表 5 中记为 JPEG+AES), 以及 JPEG 加文献 [21] 中的加密算法 (表 5 中记为 JPEG+[20]) 的实验结果, 如表 5 所示.

实验结果表明, 当压缩比较高时, 本文方法重构图像质量优于分离算法 (JPEG+AES, JPEG+[20]). 这主要是由于 JPEG 压缩中的离散余弦变化分块效应引起; 而在压缩比较低时, 本文提出的算法加密所花时间占整个算法时间的比例低于分离算法中加密时间占整个算法时间的比例. 这主要是因为分离算法需对整个压缩编码位流进行加密, 而本文的方法只根据给定的阈值选择部分系数进行加密.

5 结 论

为了保护多媒体信息内容, 本文研究了压缩之前的加密的可能性. 特别地, 在我们的算法中, 小波系数被搅乱了, 并进行了选择性地加密. 实验结果表明, 本文提出的算法实现了算法的安全性、重构图像的视觉质量以及压缩比之间的一个很好的折衷.

[1] Wen J, Kim H, Villasenor J D 2006 *IEEE trans. Signal Process Letter* **13** 69

[2] Kim H, Wen J, Villasenor J D 2007 *IEEE trans. Signal Process.* **55** 2263

[3] Bose R, Pathak A 2006 *IEEE Trans. Circuits Syst.* I **53** 848

[4] Wong K W, Yuen C H 2008 *IEEE trans. Circuits Syst.* II **55** 1193

[5] Mao Y N, Wu M 2006 *IEEE Trans, Image Processing.* **15** 2061

[6] Brown R, Chua L O 1996 *Int. J. Bifurc. Chaos* **6** 219

[7] Fridrich J 1998 *Int. J. Bifurc. Chaos* **8** 1259

[8] Jakimoski G, Kocarev L 2001 *IEEE Trans. Circuits Syst.* I **48** 163

[9] Dachselt F, Schwarz W 2001 *IEEE Trans. Circuits Syst.* I **48** 1498

[10] Said A, Pearlman W A 1996 *IEEE Trans, Image Processing* **5** 1303

[11] Shapiro J M 1993 *IEEE Trans. Signal Processing* **41** 3445

[12] Luo R C, Chung L Y, Lien C H 2002 *IEEE Trans. Industrial Electronics* **49** 933
[13] Chan K S, Fekri F 2004 *IEEE Trans. Signal Processing.* **52** 2975
[14] Lian S, Sun J, Wang Z 2004 *IEEE Int. Conf. Multimedia Exp.* **3** 2195
[15] Shannon C 1949 *Bell System Tech. J.* **28** 656
[16] Lin R, Mao Y B, Wang Z Q 2008 *Proc IEEE Int Conf. Communications and Networking in China* 1294
[17] David Salomon 2006 *Data Compression* Third Edition (Springer ISBN 0-387-40697-2)
[18] http://www.cipr.rpi.edu.research/SPIHT
[19] Kohda T, Tsuneda A 1997 *IEEE Trans Inform Theory* **43** 104
[20] Said A, Pearlman W A 1996 *IEEE Trans, Circuits Syst for Video Tech.* **6** 243
[21] Behnia A, Akhshani A, Ahadpour S, Mahmodi H, Akhavand A 2007 *Phys. Lett.* A **366** 391

SPIHT-based joint image compression and encryption*

Yang Hua-Qian[1)2)3)†] Liao Xiao-Feng[1)] Kwok-Wo Wong[2)]
Zhang Wei[3)] Wei Peng-Cheng[3)]

1) (*College of Computer Science and Engineering, Chongqing University, Chongqing* 400044*, China*)

2) (*Dept of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China*)

3) (*Centre of Network Management, Chongqing Education of College, Chongqing* 400067*, China*)

(Received 14 March 2011; revised manuscript received 20 Jane 2011)

Abstract

A joint of image encryption and compression is investigated, and a novel joint encryption and compression scheme is proposed. In our scheme, the encryption process is performed before compression. Making use of the properties of discrete wavelet transform (DWT) and Set Partitioning in Hierarchical Trees (SPIHT) coding, confusion is restricted to the interior of single subband image itself and so image details are retained. Furthermore, diffusion preserves the two most significant bits (MSBs) and the sign bit, which contain important information about the image. The experiments show that the proposed algorithm possesses a good visual quality of the reconstructed image and a high encryption speed.

Keywords: cryptography, compression, discrete wavelet transform (DWT), set partitioning in hierarchical trees (SPIHT)

PACS: 05.45.Gg

* Project supported by the Research Grants Council of the Hong Kong Government, China (Grant No. CityU 122308), the Natural Science Foundation Project of CQ CSTC (Grant No. CSTC, 2010BB2279), the Applying Basic Research Program of Chongqing Education Committee (Grant Nos. kj091501, kj091502, kj101501), and the China Postdoctoral Science Foundation (Grant No. 2011M501391).

† E-mail: mailtostorm@163.com