基于脆弱水印的图像认证算法研究

张宪海,杨永田
(哈尔滨工程大学计算机科学与技术学院,黑龙江哈尔滨 150001)

摘 要: 很多基于分块的图像认证算法为了提高安全性,采用大分块或者分块相关技术,因而牺牲了定位精度.通过对各种攻击的分析,提出了一种基于脆弱水印的图像认证方案.使用 SHA512 算法和基于背包问题的单向函数来产生水印,使用滑动窗口技术和层次结构来嵌入水印,使强加密算法在小分块上得以应用.该方案不但能够抵抗矢量量化等目前已知的各种攻击,而且能够将篡改定位到 2 ×2 大小的像素块上.理论分析和实验数据表明,该方案在保证系统安全性的同时,有效地提高了篡改定位的精度.

关键词: 脆弱水印; 层次结构; 篡改检测; 篡改定位
中图分类号: TP391 **文献标识码**: A **文章编号**: 0372-2112 (2007) 01-0034-06

Image Authentication Scheme Research Based on Fragile Watermarking

ZHANG Xian-hai, YANG Yong-tian
(School of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

Abstract: Big blocks or blocks relation techniques are applied in several image authentication schemes to improve security at the cost of localization accuracy is decreased. In this paper, an image authentication scheme based on fragile watermarking is proposed by means of the analysis of all kinds of attacks. On the basis of small blocks, the strong cryptography algorithm is applied, the watermarking is generated by SHA512 and one-way function based on knapsack problem, and is embedded via sliding window and hierarchical structure. The approach can not only resists all kinds of attacks such as vector quantization (VQ), but also localize a tamper to a 2 ×2 block pixels. Theory analysis and experimental results demonstrate that the proposed scheme improves the accuracy of tamper localization as well as ensure the security.

Key words: fragile watermarking; hierarchical structure; tamper detection; tamper localization

1 引言

多媒体技术和互联网技术的发展,使人们能够方便的获得各种多媒体信息.多媒体信息具有易于存储、发布以及二次创作等诸多优点,但同时,也存在恶意篡改等风险,在某些特殊领域,有可能造成严重后果.比如,对医学图像的篡改可能造成误诊,用于法庭证据的图像经过篡改可能扭曲真实,对指纹图像的篡改可能放过罪犯而使好人蒙冤等等,这就需要对图像的真实性(也称完整性)进行验证,以证明所需要的图像是否真实.

传统的数据认证采用基于哈希函数的消息认证码(MAC)或者基于非对称密码技术的数字签名(DS)技术,将所有数据看作二进制比特流,计算该比特流的消息认证码或数字签名,附在被保护信息尾部一起传输,或者使用另外的信道来传输.

数字图像也可以看作是比特流,使用传统的数据认证技术对图像进行认证,这种方法的缺点在于:(1)需要传输额外的数据,增加了数据量;(2)加密后的乱码容易引起攻击者的注意;(3)只能检测出是否发生篡改,无法定位篡改发生的位置,因而也就无法判断恶意攻击者的企图.基于脆弱水印的图像认证技术正好克服了上述缺点,引起很多学者的高度重视.

图像认证技术主要解决两个问题,即篡改检测和篡改定位,前者用来判断图像是否真实,后者用来判断图像的哪些地方不真实.篡改检测需要解决的主要问题是将漏警率控制在一定的水平,同时,要对抗可能出现的各种伪造真实的攻击,即算法的安全性问题;篡改定位需要解决的主要问题是准确指出篡改发生的位置,即定位精度问题.从当前的研究来看,算法的定位精度和安全性之间存在一定的矛盾.

基于水印的图像认证包括精确认证和模糊认证两类,前者不允许对图像进行哪怕是一个比特的修改;后者在保证图像语义和内容真实性的前提下,允许对图像进行压缩等正常处理操作.精确认证和模糊认证有不同的应用领域,本文关注的是图像的精确认证.

1995 年,Walton[1]首次提出用脆弱数字水印的方法实现图像认证,其主要思想是随机选择一些像素,计算他们的灰度

收稿日期:2006-05-22;修回日期:2006-09-14

值中除了最低位之外的其它位的校验和,并作为水印信号嵌入到其最低有效位(LSB)中.这篇文章是该领域最早的文献之一,虽然算法还不够安全,而且篡改定位能力比较差,但该方法原理简单,为后来的研究指明了方向.

Yeung 和 Minzter 在文献[2]中提出一种基于脆弱水印的单像素认证算法.用一个秘密函数 f 生成一个秘密 LOGO,使用错误扩散法,适当修改每个像素的灰度值嵌入水印.检测时通过比较提取的 LOGO 和最初生成的 LOGO,就可以知道哪个像素被篡改了,从而可以把篡改定位精确到一个像素.但该方案只有一半的篡改能被识别定位,其安全性严重依赖于秘密 LOGO,一旦 LOGO 和函数 f 被识别,很容易进行诸如 VQ 等攻击.而且 Fridrich[6]和 Wu 等人[7]指出不论对 Y-M 方法如何改进,只要保持这种顺序处理每个像素的方式,总可以发动 ORACLE 攻击来伪造一幅真实的图像.

1998 年,Wong 等人[3]提出一个基于脆弱水印的分块认证算法,主要思想是使用大码本和添加分块索引的方式,在各小块上嵌入各自的水印.然而,Holliman[5]指出这类分块独立算法存在致命的缺陷,并用矢量量化成功进行攻击.Wong 等人[4]后来提出了改进算法,对整个图像去掉 LSB 位进行哈希,来产生整个图像的索引,并将这个索引作为图像块哈希函数输入的一部分.这种方法可以挫败量化攻击,但却损害了定位的精度.另外,该方法使用 RSA 等强签名算法,导致图像块的大小不能小于 32 ×32 像素,因此,篡改定位最小到 32 ×32 像素块.

Fridrich 等人[6]采用分块编号和图像唯一索引来消除分块独立性,能够对抗 VQ 攻击,而且篡改定位能力更强一些,但也只能定位在 8 ×16 的分块上,而且,考虑到一个强签名算法所需要嵌入的比特数目,一个 8 ×16 分块显然是不够的.

Celik 等人[8]和 Lin 等人[9]提出了类似的基于层次结构的精确认证方法.将一幅图像分成多个层次,高一层中每个图像块由下面一层 2 ×2 共 4 个图像块组成,最高层是图像本身.因为所有的块都参与了签名,因此可以对抗任何 VQ 攻击企图.但该算法定位精度和所使用加密算法有关,如果使用 DSA 算法,最小定位到 21 ×21 像素块,而如果使用 RSA 算法,最小只能定位到 37 ×37 像素块,篡改定位能力有限.

从以上分析中可以看出,目前的图像精确认证算法可以分为单像素认证和分块认证两种[10],前者可将篡改定位在一个像素上,但容易受到 Oracle 攻击,安全性不高;后者具有较高的安全性,但只能将篡改定位在一个分块上,篡改定位能力有所下降.

本文提出了一种新的基于滑动窗口技术和层次结构安全的图像精确认证算法.该算法将篡改检测和篡改定位相分离,使用 32 ×32 的滑动窗口技术,可以对抗已知的各种攻击,并用强加密算法[11]保证系统的安全性,同时,使用层次结构进行篡改定位,将定位精确到 2 ×2 的像素块上,从而在保证安全性的前提下,尽量提高了篡改定位的精度.

本文第 2 节和第 3 节分别给出了水印生成、嵌入及篡改检测、篡改定位算法,第 4 节对算法性能进行了理论分析,第 5 节给出了实验结果,最后总结全文.

2　水印嵌入算法

为保证安全性,同时提高定位精度,本算法使用了基于背包问题的单向函数,并在密钥控制下对图像进行分块置乱变换,然后使用滑动窗口技术和层次结构生成并嵌入水印.

2.1　背包单向函数

函数 $f(x)$ 是一个单向函数是指给定 x,计算 $f(x)$ 很容易,反之,给定 $f(x)$,计算 x 却很难.本算法中使用基于背包问题的单向函数,描述如下:

给定一个含有 n 个元素的矢量 A 和一个取值为 0 或 1 的含有 n 个元素的二值矢量 B:

$$A = (a_1, a_2, \cdots, a_n) \quad \forall a_i \in N$$

$$a_1 < a_2 < \cdots < a_n, \ \sum_{k=1}^{i} a_k > a_{i+1}$$

$$i, j = 1, 2, \cdots, n$$

$$B = (b_1, b_2, \cdots, b_n), \ b_i \in \{0, 1\}$$

定义单向函数 $f(x)$ 如下:

$$f(B) = A \cdot B = \sum_{i=1}^{n} a_i * b_i \tag{1}$$

(1)从公式中我们可以看出,如果 A 和 B 已知,计算函数 $f(B)$ 将很容易,而反过来,如果 $f(B)$ 已知,想得到 B,就需要遍历矢量 A 中所有元素所有可能的组合,其遍历空间为 2^n.满足安全性要求的背包问题是一个 NP-完全问题,没有多项式时间的算法.

2.2　图像预处理

本文采用分块置乱变换技术对图像进行预处理,即首先将图像分成 4 ×4 大小的数据块,并以该数据块为最小单位进行置乱变换.这样不但可以提高系统的安全性,而且不影响篡改定位.

几何变换是常见的图像置乱技术.设像素块初始位置为 (x, y),置乱后的位置为 (x', y'),图像的大小为 $N \times N$,则置乱变换和逆置乱变换定义如公式(1)和(2)所示:

$$\begin{pmatrix} x \\ y \end{pmatrix}' = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \pmod{N/4} \tag{2}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{d}{delta} & \frac{b}{delta} \\ \frac{-c}{delta} & \frac{a}{delta} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}' \pmod{N/4} \tag{3}$$

其中,$delta = ad - bc = \pm 1$,$a, b, c, d \in N$.当 $a = b = c = 1$,$d = 2$ 时,就是著名的 Arnold 变换;当 $a = b = c = 1$,$d = 0$ 时,就是 Fibonacci 变换.图 1 显示的是 Lena 原始灰度图像及其在 $a = 7$、$b = 11$、$c = 5$、$d = 8$ 情况下,迭代 35 次置乱后的图像.

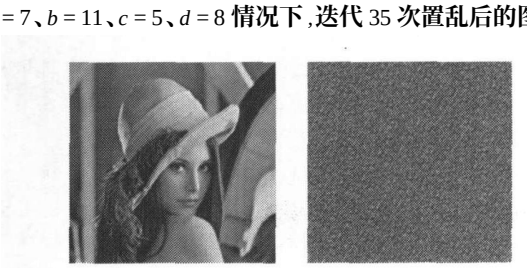


图 1　原始图像和分块置乱后的图像

从图 1 可以直观地看出,分块置乱和像素级置乱一样,都可以达到很好的置乱效果.

2.3 水印生成和嵌入

本文提出了一种基于滑动窗口技术和层次结构的水印生成及嵌入算法. 设原始图像 I 的大小为 $M \times N$, M 和 N 是 8 的整数倍(如果不是则要补足),设分块置乱后的图像为 I_p,则水印的生成和嵌入在图像 I_p 上进行,嵌入结束通过逆图像分块置乱,得到嵌入水印后的图像 I'.

图 2 给出了滑动窗口和层次结构示意图. 其中,符号 △、符号 ☆和符号 # 分别代表了 4 ×4 像素块水印、2 ×2 像素块水印和整个滑动窗口水印.

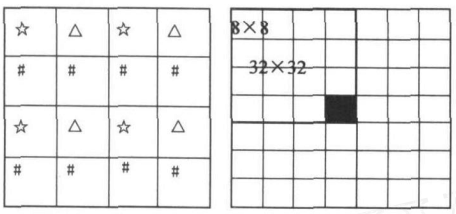


图 2 水印层次结构和滑动窗口示意图

水印生成及嵌入过程如图 3 所示,具体步骤如下:

Step 1. 用置乱变换密钥 $k_{\text{permutation}}$ 和迭代次数 k_n 对图像 I 进行分块置乱变换,得到分块置乱后的图像 I_p.

Step 2. 将图像 I_p 所有像素的 LSB 位置 0.

Step 3. 使用一个 32 ×32 大小的滑动窗口,从图像左上角开始,按照从左到右,从上到下的顺序扫描整个图像,窗口每一次滑动的步长是 8 个像素行或列. 当滑动窗口到达图像的最右边或最下边时,循环使用其最左边和最上边的像素行或列. 这样,任何一个 8 ×8 大小的像素块将被 16 个滑动窗口所扫描(如图 2 所示). 将所有滑动窗口按照从左到右,从上到下的顺序依次编号为 $(1,1)$, $(1,2)$, …, $(M/8, N/8)$;

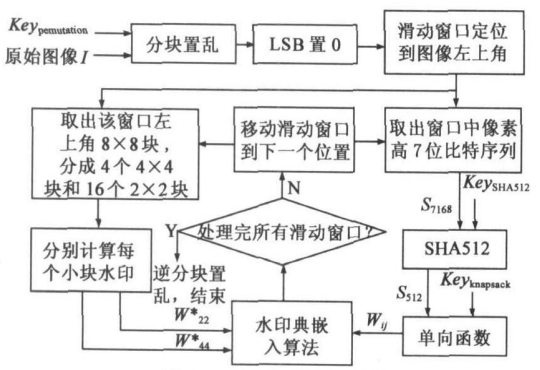


图 3 水印生成嵌入算法

Step 4. 对第 (i,j) 个滑动窗口,进行如下操作:

1. 将窗口中所有像素按行扫描顺序依次排列,并提取出每个像素的高 7 比特位,组成长度为 7168 的比特序列 $S_{i,j}$:

$$S_{i,j} = \{ s^1_{i,j}, s^2_{i,j}, \cdots, s^{7168}_{i,j} \}, \quad s^r_{i,j} \in \{0,1\}$$

2. 使用安全哈希函数 SHA512[11] 对比特序列 $S_{i,j}$ 进行哈希运算,得到长度为 512 比特的滑动窗口信息摘要:

$$M_{i,j} = \{ m^1_{i,j}, m^2_{i,j}, \cdots, m^{512}_{i,j} \}, \quad m^r_{i,j} \in \{0,1\}$$

3. 使用基于背包问题的单向函数 f,将 512 比特消息摘要映射成 32 比特. 在密钥控制下,取一个长度为 512 元素,满足背包问题安全性要求的伪随机正整数矢量 A,并计算和值 $\tilde{W}_{i,j}$:

$$A = (a_1, a_2, \cdots, a_{512}), \quad a_i \in N, \ a_i < 8388608$$

$$\tilde{W}_{i,j} = f(M_{i,j}) = \sum_{r=1}^{512} a_r * m^r_{i,j} \tag{4}$$

4. 将 $\tilde{W}_{i,j}$ 写成二进制序列,如果序列长度不足 32 比特,在前面补上 0,得到长度为 32 比特的第 (i,j) 个滑动窗口的水印信息:

$$W_{i,j} = \{ w^1_{i,j}, w^2_{i,j}, \cdots, w^{32}_{i,j} \}, \quad w^r_{i,j} \in \{0,1\}$$

Step 5. 取出该滑动窗口左上角的 8 ×8 小块,划分成 4 个 4 ×4 小块. 取出其中的一个 4 ×4 小块,计算并嵌入篡改定位水印信息,方法如下:

1. 取出一个 4 ×4 小块 $\tilde{B}_{44}(i,j)$, $i,j \in \{1,2,3,4\}$,计算该小块的块均值 μ_{44};

$$\mu_{44} = \sum_{i=1}^{4} \sum_{j=1}^{4} \tilde{B}_{44}(i,j) / 16 \tag{5}$$

2. 将该小块映射成二值图像块:

$$B_{44}(i,j) = \begin{cases} 1 & \text{if} \quad \tilde{B}_{44}(i,j) \geq \mu_{44} \\ 0 & \text{otherwise} \end{cases} \quad i,j \in \{1,2,3,4\} \tag{6}$$

3. 计算得到该二值图像块行、列二值序列:

$$B\_row(i) = \begin{cases} 1 & \text{if} \quad \sum_{j=1}^{4} \tilde{B}_{44}(i,j) > 2 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$B\_col(j) = \begin{cases} 1 & \text{if} \quad \sum_{i=1}^{4} \tilde{B}_{44}(i,j) > 2 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

4. 计算图像块 $\tilde{B}_{44}(i,j)$ 的水印信息:

$$W_{44}(i) = B\_row(i) \oplus B\_col(i), \quad i = 1, \cdots, 4$$

5. 将 4 比特的水印信息分别替换像素 $\tilde{B}_{44}(1,2)$、$\tilde{B}_{44}(1,4)$、$\tilde{B}_{44}(3,2)$、$\tilde{B}_{44}(3,4)$ 的 LSB 位,即图 2 中符号 △所指示的像素的位置;

6. 将每个 4 ×4 小块 $\tilde{B}_{44}(i,j)$, $i,j \in \{1,2,3,4\}$ 划分成 4 个 2 ×2 小块 $\tilde{B}_{22}(i',j')$, $i',j' \in \{1,2\}$,计算该 2 ×2 小块的块均值:

$$\mu_{22} = \sum_{i'=1}^{2} \sum_{j'=1}^{2} \tilde{B}_{22}(i',j') \tag{9}$$

7. 计算块 2 ×2 小块 $\tilde{B}_{22}(i',j')$ 的水印:

$$W_{22} = \begin{cases} 1 & \text{if} \quad \text{num}(\mu_{22}) \quad \text{is odd} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

其中 $\text{num}(\mu_{22})$ 是指块均值 μ_{22} 中取值为 1 的比特的个数;

8. 将水印 W_{22} 嵌入到像素 $\tilde{B}_{22}(1,1)$ 的 LSB 位置上,即图 2 中符号 ☆所指示的像素的位置;

9. 取出下一个 2 ×2 小块,重复上述 7 到 8 步,直到 4 个 2 ×2 小块都嵌入篡改定位水印;

10. 取出下一个 4 ×4 小块,重复上述 1 到 9 步,直到 4 个 4 ×4 小块都嵌入篡改定位水印;

Step 6. 将第 (i,j) 个滑动窗口的水印信息 $W_{i,j}$ 嵌入到该滑动窗口左上角的 8 ×8 小块偶数行像素的 LSB 位置上;

Step 7. 将滑动窗口以步长为 8 个像素滑动到下一个位置,重复上述 Step 4 至 Step 6 的操作,将水印嵌入到所有滑动窗口中;

Setp 8. 分块图像逆置乱变换,得到嵌入水印后图像 I'.

3 篡改检测和篡改定位

在检测端包括参考水印计算、嵌入水印提取、篡改检测和篡改定位等几个部分,检测时不需要原始图像,只需要知道嵌入密钥即可,是完全的盲提取算法.

3.1 篡改检测

设 I 为待检测图像,篡改检测过程如图 4 所示. 具体过程如下:

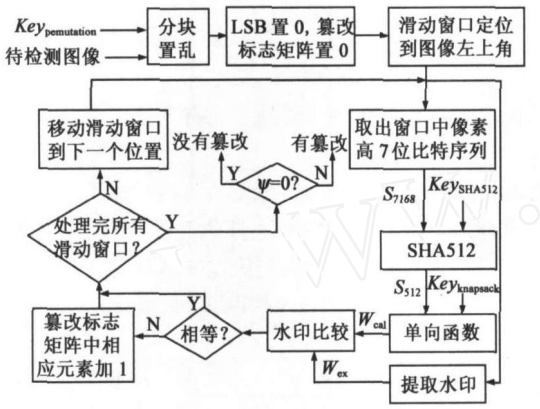


图 4　篡改检测算法

Step 1. 用相同的密钥对图像 I 进行分块置乱变换,得到分块置乱后的图像 I_p'.

Step 2. 生成一个正整数矩阵 Ψ 并将该矩阵所有元素值初始化为 0,该矩阵中元素的位置和滑动窗口的位置是一一对应的;

$$\Psi(i,j)=0;\ i=1,\cdots,M/8;\ j=1,\cdots,N/8$$

Step 3. 按照 2.2 节的方法计算第 (i,j) 个滑动窗口的水印信息 $W_{cal_{i,j}}$,并提取第 (i,j) 个滑动窗口的水印信息 $W_{ex_{i,j}}$;

Step 4. 按照比特顺序依次比较 $W_{cal_{i,j}}$ 和 $W_{ex_{i,j}}$ 的每一个比特,并计算 Ψ,如公式(11)所示;

$$\Psi(i+i_1,j+j_1)=\begin{cases}\Psi(i+i_1,j+j_1)+1 & \text{if } W_{cal_{i,j}}\neq W_{ex_{i,j}}\\ \Psi(i+i_1,j+j_1) & \text{otherwise}\end{cases}$$
$$i_1=0,\cdots,3;\ j_1=0,\cdots,3 \quad (11)$$

Step 5. 对所有滑动窗口依次执行上述 Step 3 至 Step 4 操作,得到篡改标志矩阵 Ψ,并输出篡改检测结果:

$$V_{tamper}=\begin{cases}\text{true} & \text{if } \sum_{i=1}^{M/8}\sum_{j=1}^{N/8}\Psi(i,j)\neq 0\\ \text{false} & \text{otherwise}\end{cases} \quad (12)$$

Step 6. 如果 $V_{tamper}=\text{false}$ 则说明被检测图像通过检测,没有任何篡改发生,检测过程结束;否则,执行下面的操作进行精细的篡改定位.

3.2 篡改定位

如果检测到有篡改发生,则需要进行篡改定位. 本算法在初步定位的 8×8 块的基础上,使用分层水印即 4×4 块和 2×2 块的水印来实现精确定位,篡改定位过程如图 5 所示,具体如下:

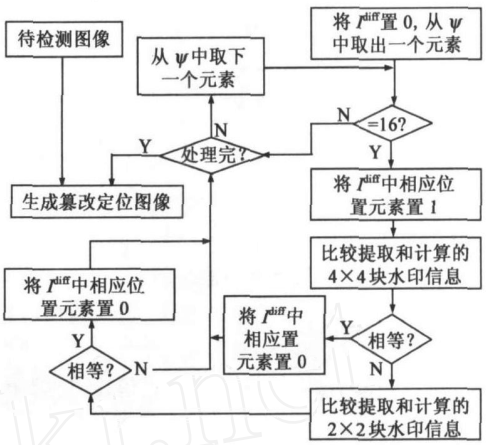


图 5　篡改定位算法

Step 1. 生成一个与被检测图像大小一致的篡改定位图像 I^{diff},初始化所有值为 0(黑色).

Step 2. 依次扫描篡改标志矩阵 Ψ 的每一个元素. 对于第 (i,j) 个元素,如果 $\Psi\neq 16$,则说明被检测图像的第 (i,j) 个 8×8 像素块通过检测,继续扫描下一个元素;否则,说明该 8×8 像素块被篡改,将 $I^{diff}(M,N)$ 中对应该位置的所有元素置 255(白色),提取出该图像块,执行下面的操作进行精细篡改定位;

Step 3. 将该 8×8 像素块分成 4 个 4×4 子块并依次处理,比较每一个子块的提取水印 $W_{ex_{44}}$ 和计算水印 $W_{cal_{44}}$,如果相等,将 $I^{diff}(M,N)$ 中对应 4×4 子块位置元素置 0,否则,则说明该子块有篡改发生. 进一步将该 4×4 子块分成 4 个 2×2 子块并依次处理,比较每一个子块的提取水印 $W_{ex_{44}}$ 和计算水印 $W_{cal_{44}}$,如果相等,将 $I^{diff}(M,N)$ 中对应 2×2 子块位置元素置 0. 如果判断有篡改发生的 4×4 块中所有 4 个 2×2 子块都没有识别出有篡改发生,则将整个 4×4 块位置元素设置为 255;

Step 4. 依次对所有图像块进行上述 Step 2 和 Step 3 的处理工作,最后,生成篡改定位图像 $I^{diff}(M,N)$;

Step 5. 对篡改定位图像 $I^{diff}(M,N)$ 进行逆置乱变换,生成带有直观篡改定位结果图像.

4 性能分析

4.1 图像质量分析

嵌入水印后的图像质量用峰值信噪比(PSNR)来衡量,设原始灰度图像为 I,嵌入水印后的图像为 I',图像大小为 $M\times N$,则 PSNR 和 MSE 定义如下:

$$PSNR=10\times\log_{10}\frac{255^2}{MSE} \quad (13)$$

$$MSE=\frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I'(i,j)-I(i,j))^2}{M\times N} \quad (14)$$

在所有 $M \times N$ 个像素中,因为 SHA512 和单向函数输出序列的伪随机性,每个像素被改变的概率为 1/2,所以平均情况下,有 $(M \times N)/2$ 个像素改变了最低比特位,所以有如下不等式:

$$MSE \leq \frac{\frac{1}{2}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(1)^2}{M \times N} = \frac{1}{2} \tag{15}$$

$$PSNR = 10 \times \log_{10}\frac{255^2}{MSE} \geq 10 \times \log_{10}\frac{255^2}{1/2} = 51.1 \text{ dB} \tag{16}$$

即便在最坏情况下,嵌入水印的图像也能达到 48dB,完全符合嵌入水印图像质量要求.

4.2 篡改检测虚警概率和漏警概率分析

虚警概率是指实际上没有发生篡改,检测器却报告有篡改发生. 根据算法可知,当输入比特不变的情况下,计算水印将和提取的水印完全相同,不可能出现错误输出,因此,该算法的虚警概率 $P_{FA} = 0$.

漏警概率是指发生了篡改,检测器却报告没有发生. 对某一个检测单元,设检测器生成的认证比特为 $W_1(i)$,$(i = 1,2,\cdots,p)$,提取的认证比特为 $W_2(i)$,$(i = 1,2,\cdots,p)$. 根据哈希函数和单向函数的特性,$W_1(i)$ 和 $W_2(i)$ 均可以看成是二值随机变量,则 $W_1(i)$ 和 $W_2(i)$ 相同的概率为 1/2. 假设共有 β 个认证单元被篡改,则漏警概率为

$$P_M = 2^{-p \cdot \beta} \tag{17}$$

因此,本算法中,单独 8 ×8 像素块的漏警概率为 $1/2^{32}$,单独 4 ×4 块的漏警概率约等于 $1/2^4$;单独 2 ×2 块的漏警概率约等于 1/2,完全可以检测出恶意篡改,因为对恶意篡改来说,改变的太少是无法达到目的的.

4.3 安全性分析

本算法首先使用密钥控制下的几何变换对图像进行分块置乱处理,由于密钥和迭代次数的空间足够大,因此,破译者必须尝试所有的置乱可能,也就是所有小图像块的全排列 $(M \times N/16)!$,图像越大,需要遍历的空间越大,破解的难度越大. 比如一个 512 ×512 大小的图像,其遍历空间将达到 $(2^{14})!$.

水印的生成使用了 SHA512 哈希运算和基于背包问题的单向函数. SHA512 哈希运算能够将任何长度小于 2^{128} 比特输入序列变换为长度为 512 比特的输出序列. 该算法输入序列中只要有一个比特的改变,在输出序列中将有一半左右的比特改变,而且该算法出现碰撞的概率极低,在可计算的时间内是无法破解的;背包问题是典型的 NP 完全问题,没有多项式时间内的算法. 两种加密算法的联合使用,既保证了水印的安全性,又减小了水印的长度. 这样就成功的在篡改检测和篡改定位之间进行了折衷处理,在保证系统安全性的同时,有效地提高了篡改定位的精度.

按照 J. Fridrich 的观点[6],用于图像认证的脆弱水印主要面临三类安全问题即检测不出来的篡改、信息泄漏和协议弱点,主要攻击方法有量化攻击(VQ)、黑盒攻击(Oracle)、密码分析攻击和特征选取攻击. 其中,特征选取攻击主要针对半脆弱水印系统,对密码分析攻击可以通过使用强加密算法如 SHA512 等并在不同图像中使用不同的密钥来防范,对单像素认证算法威胁最大的是黑盒攻击,对分块认证算法威胁最大的是量化攻击.

本算法中使用图像分块置乱技术和滑动窗口技术有效地抵抗矢量量化攻击. 本算法首先对原始图像进行分块置乱变换,攻击者无法找到一个合适的、而且在置乱后正好能组合成一个完整的 8 ×8 大小的替换块;而且认证图像中任何一个 8 ×8 小块都和其四周的 49 个 8 ×8 小块相关联,和 16 个滑动窗口相关,整个图像中任何一个 8 ×8 小块都不是孤立的,替换一个小块将引起其他小块的认证失败,因此,本算法完全可以对抗矢量量化攻击.

从以上分析中可以看出,本文算法可以抵抗到目前为止的所有已知攻击.

5 实验结果

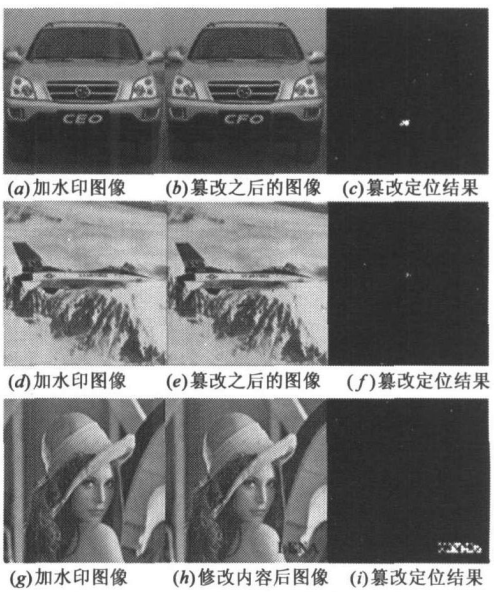
本实验以标准 512 ×512 的 8 比特灰度图像来测试算法性能.

表 1 给出了不同图像嵌入水印后的 PSNR 值,从表中可以看出,所测试图像的峰值信噪比都达到了 51dB 以上,和理论分析是一致的,说明本算法具有极佳的不可见性.

表 1 图像嵌入水印后的 PSNR 值

Image	Lena	Peppers	Boat	Car	F16
PSNR	51.14	51.76	51.90	51.85	51.70

图 6(a)、(b)和(c)分别给出了加水印之后的汽车图像、将字母"CEO"改成"CFO"后的图像以及篡改定位结果;图 6(d)、(e)和(f)分别给出了加水印之后的 F16 图像、将字母"F16"改成"F18",同时将字母"US"改成"UN"后的图像以及篡改定位结果;图 6(g)、(h)和(i)给出了加水印之后的 Lena 图像、在右下角加上文字之后的图像及其篡改定位输出结果. 从图中我们可以看出,文本算法在篡改定位方面具有很好的精度,可以准确的指出篡改发生的区间和位置.



(a)加水印图像 (b)篡改之后的图像 (c)篡改定位结果

(d)加水印图像 (e)篡改之后的图像 (f)篡改定位结果

(g)加水印图像 (h)修改内容后图像 (i)篡改定位结果

图 6 篡改之后的图像及篡改检测结果

6 结论

本文提出了一种使用滑动窗口技术和层次结构的图像精确认证算法. 该算法使用了 SHA512 强加密算法和基于背包问题的单向函数,有效地保证了系统的安全性. 分块置乱变换和滑动窗口的使用使得算法有效第抵抗矢量量化攻击,同时,层次结构的篡改定位技术将定位精度提高到了 2 ×2 大小的像素块. 实验结果表明所提出的算法既能准确检测篡改,又能精确定位篡改,从而验证了理论分析的正确性. 在检测端不需要原始图像,只需要知道密钥就可以实现篡改检测和定位,是一个安全且实用的图像认证算法.

本文算法的主要思想能够很容易的应用到其他基于分块的图像认证算法中,以用来抵抗矢量量化攻击. 在计算时间允许的情况下,该算法还可以结合 RSA1024 算法一起使用,即将 32 个滑动窗口的水印序列进一步用 RSA1024 算法进行加密处理,然后一起嵌入,从而进一步提高系统的安全性.

参考文献:

[1] Walton S. Image authentication for a slippery new age[J]. Dr. Dobb's Journal,1995,20(4):18 - 26.

[2] Yeung M, Mintzer F. An invisible watermarking technique for image verification[A]. In Proceedings of the IEEE International Conference on Image Processing[C]. Santa Barbara, USA, 1997. 680 - 683.

[3] Wong P W. A public key watermark for image verification and authentication[A]. In Proceedings of the IEEE International Conference on Image Processing[C]. Chicago, USA, 1998. 455 - 459.

[4] Wong P W, Memon N. Secret and public key image watermarking schemes for image authentication and ownership verification [J]. IEEE Transactions on Image Processing, 2001, 10(10): 1593 - 1601.

[5] Holliman M, Memon N. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes[J]. IEEE Transaction on Image Processing, 2000, 9(3): 432 - 441.

[6] Fridrich J. Security of fragile authentication watermarks with localization[A]. In Proceedings of the SPIE Security and Watermarking of Multimedia Contents IV[C]. San Jose, California, 2002. 691 - 700.

[7] Wu Jin-Hai, Zhu Bin, Lin Fu-Zong. Efficient oracle attacks on Yeung-Mintzer and variant authentication schemes[A]. In Proceedings of the IEEE International Conference on Multimedia & Expo (ICME '04)[C]. Taiwan, 2004. 931 - 934.

[8] Celik M, Sharma G. Hierarchical watermarking for secure image authentication with localization[J]. IEEE Transactions on Image Processing, 2002, 11(6): 585 - 594.

[9] Lin P, Hsieh C. A hierarchical watermarking method for image tamper detection and recovery[J]. Pattern Recognition, 2005, 38(2): 2519 - 2529.

[10] 吴金海,林福宗. 基于数字水印的图像认证技术[J]. 计算机学报,2004,27(9):1153 - 1161.
Wu Jin-Hai, Lin Fu-Zong. Image authentication based on digital watermarking[J]. Chinese journal of computers, 2004, 27(9): 1153-1161. (in Chinese)

[11] National Institute of Standards and Technology NIST. Secure hash standard[S]. Federal Information Processing Standards Publication 180 - 2, 2002.

作者简介:

张宪海　男,1971 年 5 月出生于吉林省蛟河市,1993 年和 2001 年分别在解放军信息工程大学和解放军国防科技大学获工学学士和工学硕士学位,现为哈尔滨工程大学计算机科学与技术学院博士研究生,主要研究方向为信息安全、数字水印、图像处理.
E-mail: xianhaizh@sina.com

杨永田　男,1939 年 12 月出生于黑龙江省哈尔滨市,1966 年毕业于北京大学计算技术专业,1966 年至 1985 年在中国航空计算技术研究所工作,期间曾在英国克兰菲尔德理工大学作访问学者,现为哈尔滨工程大学教授,博士生导师,主要研究方向为计算机网络及应用、分布式计算系统、信息安全和仿真技术等. 已出版著作 4 部,发表论文 60 余篇.