基于矩阵编码和多重水印的JPEG图像块级认证算法

巩道福[1,2]，刘粉林[1,2]，罗向阳[1,2,3]，汪 萍[4]

(1. 解放军信息工程大学四院 郑州 450001；2. 数学工程与先进计算国家重点实验室 郑州 450001;
3. 中国科学院信息工程研究所信息安全国家重点实验室 北京 海淀区 100093；4. 河南省臻嘉科技有限公司 郑州 450002)

【摘要】JPEG图像是目前常用的图像格式之一，由于其冗余数据较少，因此实现对JPEG图像的精确认证，需要对认证水印信息进行合适的编码以尽可能地缩短水印长度。为此，结合矩阵编码和多重水印技术，提出了一种用于JPEG图像的块级认证算法。矩阵编码可将长为l的水印信息缩短为$\log_2 l$数量级，且可准确定位其中1位信息的改变，因此通过矩阵编码可有效缩短水印的长度。同时，为了提高算法的定位准确率，通过嵌入多重水印共同完成对图像的篡改定位。算法分析中对算法的漏检率、虚检率进行了较为详细的分析。理论分析与实验结果表明，在篡改率不大的情况下，算法能够实现对JPEG图像的块级篡改定位，在篡改率小于5%，向每个图像块仅嵌入约2 bit水印信息的情况下，漏检率和虚检率均在10^{-2}数量级。

关 键 词 图像认证； JPEG图像； 矩阵编码； 多重水印
中图分类号 TP391 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.03.019

Block Level Authentication Algorithm for JPEG Image Based on Matrix Coding and Multi-Watermarking

GONG Dao-fu[1,2], LIU Fen-lin[1,2], LUO Xiang-yang[1,2,3], and WANG Ping[4]

(1. The fourth Institute of PLA Information Engineering University Zhengzhou 450001;
2. State Key Laboratory of Mathematical Engineering and Advanced Computing Zhengzhou 450001;
3. State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences Haidian Beijing 100093;
4. Vega Info & Tech Co., Ltd. of Henan Province Zhengzhou 450002)

Abstract JPEG image is one of the commonly used image format. Because of the less redundant data, the length of authentication watermark for JPEG images authentication should be as short as possible. In this paper, combined with matrix encoding and multiple watermarking, a block-level authentication algorithm for JPEG images is proposed. Matrix encoding can shorten the length of the watermark magnitude and accurately locate one information change. And then multi-watermarks are generated for the tampering of the image block authentication. The detailed analysis is given about the missing rate and the false detection rate in this paper. Theoretical analysis and experimental results show that the JPEG images block-level tamper localization can be done under the lower tamper rate. The missing rate and the false detection rate are at the 10^{-2} level, even if 2 bits watermark are embedded in the rate of tampering is less than 5%.

Key words image authentication; JPEG image; matrix encoding; multi-watermark

数字图像容易被修改的特点带来了一系列的安全隐患，使得使用者难以判断其内容的真伪。一些个人或团体为达到某种目的，对图像进行非法篡改并恶意传播，不仅会给社会造成不良的影响，甚至导致重大的经济或政治损失。因此，如何对数字图像进行可靠认证，包括真实性鉴别、定位被篡改区域等，是目前信息安全领域急需解决的问题之一。

图像认证技术的研究主要集中在空域图像认证和JPEG图像认证。目前，在空域图像方面的认证技术取得了一定的研究成果[1-5]，但针对JPEG图像的认证算法较少，主要是因为JPEG图像以量化后的DCT系数的形式进行存储，其中大部分系数为0，数据冗余较少，难以像空域图像一样通过嵌入大量的信息进行准确定位甚至恢复[6]。目前，针对JPEG图像的认证算法主要可分为两类：一类是结合JPEG压缩不变性的半脆弱水印算法[7-8]，其按照一个较大的量化表对载体图像进行压缩处理，该处理过程会对载体图像的质量产生较大影响，固水印的不可见性较差。

收稿日期：2013 – 01 –10；修回日期：2013 – 06 – 18
基金项目：国家自然科学基金(61272489, 61379151, 61302159)；河南省科技攻关计划(122102210516)；中国博士后基金(2012T50842, 201104911838)
作者简介：巩道福(1984 –)，男，博士，主要从事网络信息安全、数字水印方面的研究.

另一类是直接在JPEG图像量化后的DCT系数上嵌入水印[6,9]，该类算法不改变原始载体图像的压缩质量因子，对图像质量影响较小，但由于数据冗余较少，认证的精度有待提高。

本文结合矩阵编码和多重水印提出一种JPEG图像块级认证算法，水印信息直接嵌入在量化后的DCT系数上。首先对原始图像进行块置乱，对置乱后的图像块进行分组，对于每一个分组使用矩阵编码的形式生成水印，可有效缩小水印长度，并通过嵌入多重水印来共同对篡改的图像块进行认证。

1　算法使用到的相关定义

设原始JPEG图像为 $\boldsymbol{X}=(x_1,x_2,\cdots,x_r)$，$\boldsymbol{x}_i$ 为由量化后的DCT系数组成的8×8的图像块，r 为图像块的总个数。对每个块的DCT系数按照一定的顺序进行排列后组成一个序列，则图像$\boldsymbol{X}$也可表示为：

$$\boldsymbol{X}=((c_1,c_2,\cdots,c_{64})_1,(c_1,c_2,\cdots,c_{64})_2,\cdots,(c_1,c_2,\cdots,c_{64})_r) \tag{1}$$

式中，$\boldsymbol{x}_i=(c_1,c_2,\cdots,c_{64})_i$ 为$\boldsymbol{X}$的第 i 个图像块；c_j 为其第 j 个量化后的DCT系数，$i=1,2,\cdots,r$；$j=1,2,\cdots,64$。

为方便算法的描述，首先给出如下定义：

定义 1　对于图像 $\boldsymbol{X}=((c_1,c_2,\cdots,c_{64})_1,(c_1,c_2,\cdots,c_{64})_2,\cdots,(c_1,c_2,\cdots,c_{64})_r)$，定义 $\boldsymbol{C}^w=(c_1^w,c_2^w,\cdots,c_m^w)=\zeta(\boldsymbol{X},K_0)$ 为水印嵌入点选取函数，其中 $c_i^w\in\boldsymbol{X}$，$i=1,2,\cdots,m$，K_0 为选取密钥。具体方法为使用 K_0 伪随即地从$\boldsymbol{X}$的非零元素中选取 m 个作为水印嵌入点。

定义 2　对于图像$\boldsymbol{X}$，定义 $\hat{\boldsymbol{X}}=\vartheta(\boldsymbol{X},\boldsymbol{C}^w)$ 为对图像进行偶数化处理，其中 $\boldsymbol{C}^w$ 为水印嵌入点序列，$\hat{\boldsymbol{X}}$ 为偶数化处理后的图像。具体方法为：对于$\boldsymbol{X}$中被选为水印嵌入点的DCT系数 $c_i^w\ (i=1,2,\cdots,m)$进行如下处理：

$$\hat{c}_i^w=\begin{cases} c_i^w & c_i^w\text{为偶数} \\ c_i^w+\mathrm{sign}(c_i^w) & c_i^w\text{为奇数} \end{cases} \tag{2}$$

即若 c_i^w 为偶数则保持不变，若为奇数则保持符号不变，绝对值加1，其中 $\hat{c}_i^w$ 为偶数化后的DCT系数。

定义 3　对于图像 $\hat{\boldsymbol{X}}$，定义 $\hat{\boldsymbol{X}}_i^b=\xi(\hat{\boldsymbol{X}},K_i^b)$ 为块置乱函数，其中 K_i^b 为置乱密钥，$\hat{\boldsymbol{X}}_i^b$ 为块置乱后的图像，$i=1,2,\cdots,k$。

定义 4　对于置乱后的图像 $\hat{\boldsymbol{X}}_i^b$，定义 $\boldsymbol{W}_i=(W_i^1,W_i^2,\cdots,W_i^p)=\psi(\hat{\boldsymbol{X}}^b,K_i^p)$ 为水印生成函数，其中 K_i^p 为水印生成密钥，$i=1,2,\cdots,k$。具体过程为：

1) 对 $\hat{\boldsymbol{X}}_i^b$ 中图像块按顺序进行无重叠分组，使每组包含 l 个图像块，共分为 $g=\lceil r/l\rceil$ 组，若 r 不为 l 的整数倍，添0补足，则第 j 个分组可表示为 $\boldsymbol{G}_j=(\hat{x}_{(j-1)\times l+1},\hat{x}_{(j-1)\times l+2},\cdots,\hat{x}_{j\times l})$，$j=1,2,\cdots,g$。

2) 对于每一个分组 $\boldsymbol{G}_j$，分别将其中每个块的DCT系数进行行优先排列并输入一个带密钥的Hash函数 H 进行运算，得到128 bit的散列值，表示为：$h_s^{128}=H(\hat{x}_s^p,K_i^p)$，其中 K_i^p 为水印生成密钥，则对于图像组 $\boldsymbol{G}_j$ 可生成 l 个128 bit的散列值，表示为：$\boldsymbol{h}_G=(h_1^{128},h_2^{128},\cdots,h_l^{128})$。

3) 将每一个 h_s^{128} ($s=1,2,\cdots,l$)中的所有比特进行异或运算，得到1 bit的信息，表示为 $b_s=\oplus(h_s^{128})$，则对于图像组 $\hat{\boldsymbol{G}}^p$ 可生成 l bit的信息，表示为 $\boldsymbol{b}_G=(b_1,b_2,\cdots,b_l)$。

4) 对 $\boldsymbol{b}_G$ 进行矩阵编码，生成 n bit的水印信息 $\boldsymbol{W}_i^p=(w_1,w_2,\cdots,w_n)$：

$$\boldsymbol{W}_i^p=\bigoplus_{j=1}^{n}b_j\times j \tag{3}$$

式中，j 在运算过程中用二进制表示，则生成的水印长度 n 需满足 $n=\lceil\log_2(l+1)\rceil$。

5) 当所有分组处理完毕后，得到 $n\times\lceil r/l\rceil$ bit的水印信息 $\boldsymbol{W}_i$。

定义 5　对于偶数化后的水印嵌入点 $\hat{c}_i^w$ 和需嵌入的水印分量 $w_i\in\{0,1\}$，定义 $\tilde{c}_i^w=\varepsilon(\hat{c}_i^w,w_i)$ 为水印嵌入函数，具体的嵌入方法为：

$$\tilde{c}_i^w=\varepsilon(\hat{c}_i^w,w_i)=\begin{cases} \hat{c}_i^w & w_i=0 \\ \hat{c}_i^w-\mathrm{sign}(\hat{c}_i^w) & w_i=1 \end{cases} \tag{4}$$

定义 6　对于嵌入水印后的DCT系数 $\tilde{c}_i^w$，定义 $w_i=\eta(\tilde{c}_i^w)$ 为水印提取函数，具体的提取方法为：

$$w_i=\eta(\tilde{c}_i^w)=\begin{cases} 0 & \tilde{c}_i^w\text{为偶数} \\ 1 & \tilde{c}_i^w\text{为奇数} \end{cases} \tag{5}$$

2　算法描述

2.1　水印生成及嵌入

设嵌入 k 重水印信息，水印生成及嵌入过程如图1所示。具体步骤可描述为：

1) 在密钥 K_0 的控制下，使用水印嵌入点选取函数 $\boldsymbol{C}^w=\zeta(\boldsymbol{X}_c,K_0)=(c_1^w,c_2^w,\cdots,c_m^w)$，选取原始图像的 m 个非零DCT系数作为水印嵌入点。

2) 使用偶数化函数 $\hat{\boldsymbol{X}}=\vartheta(\boldsymbol{X},\boldsymbol{C}^w)$ 对图像$\boldsymbol{X}$进行偶数化处理。

3) 分别在密钥 $K_1^b,K_2^b,\cdots,K_k^b$ 的控制下，使用块置乱函数 $\hat{\boldsymbol{X}}_i^b=\xi(\hat{\boldsymbol{X}},K_i^b)$ 对图像 $\hat{\boldsymbol{X}}$ 进行置乱处理，得

到 k 副置乱后的图像 $\hat{\boldsymbol{X}}_1^b,\hat{\boldsymbol{X}}_2^b,\cdots,\hat{\boldsymbol{X}}_k^b$。

4) 分别在密钥 $K_1^p,K_2^p,\cdots,K_k^p$ 的控制下，使用水印生成函数 $\boldsymbol{W}_i=\psi(\hat{\boldsymbol{X}}^b,K_i^p)$，对置乱后的图像 $\hat{\boldsymbol{X}}_1^b,\hat{\boldsymbol{X}}_2^b,\cdots,\hat{\boldsymbol{X}}_k^b$ 生成水印信息，共生成 k 个水印信息，并进行串联作为整幅图像的水印信息，表示为 $\boldsymbol{W}=(w_1,w_2,\cdots,w_m)=(W_1,W_2,\cdots,W_k)$，其中，$m=n\times\lceil r/l\rceil\times k=\lceil\log_2(l+1)\rceil\times\lceil r/l\rceil\times k$ 为水印信息的总长度。

5) 使用水印嵌入函数 $\varepsilon(\hat{c}_i^w,w_i)$ 将生成的水印嵌入在所选取的DCT系数上，生成含水印图像 $\boldsymbol{X}^w$。

本文水印算法共生成 k 重水印信息，在算法分析中，将给出该设计的考虑。同时，步骤1)中选取的嵌入点个数m与步骤5)中生成的水印信息长度 m 为相同的值，因此在选取嵌入点之前，应首先根据需生成的水印组数 k 以及对图像块进行分组的大小 l，确定水印信息的长度 m。

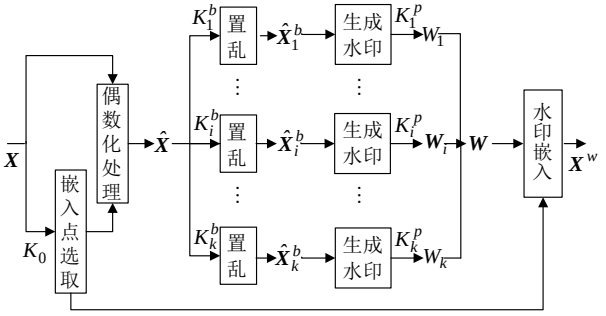


图1 水印生成及嵌入流程图

2.2 篡改检测及定位

设 $\tilde{\boldsymbol{X}}=(\tilde{x}_1,\tilde{x}_2,\cdots,\tilde{x}_r)$ 为可能遭受篡改的含水印图像，其篡改检测过程如图2所示，具体算法步骤为：

1) 在密钥 K_0 的控制下，使用水印嵌入点选取函数 $\boldsymbol{C}^w=\zeta(\boldsymbol{X}_c,K_0)=(c_1^w,c_2^w,\cdots,c_m^w)$，选取图像 $\tilde{\boldsymbol{X}}$ 的 m 个非零DCT系数。

2) 由水印提取函数 $w_i=\eta(\tilde{c}_i^w)$，从选取的DCT系数中提取所嵌入的水印信息 $\tilde{\boldsymbol{W}}=(\tilde{w}_1,\tilde{w}_2,\cdots,\tilde{w}_m)$。

3) 对待检测图像 $\tilde{\boldsymbol{X}}$，在密钥 $K_1^b,K_2^b,\cdots,K_k^b$、$K_1^p,K_2^p,\cdots,K_k^p$ 的控制下，由3.1节中的方法生成参考水印 $\bar{\boldsymbol{W}}=(\bar{w}_1,\bar{w}_2,\cdots,\bar{w}_m)$。

4) 比较 $\tilde{\boldsymbol{W}}$ 和 $\bar{\boldsymbol{W}}$ 确定被篡改图像块，具体为：

① 将 $\tilde{\boldsymbol{W}}$ 和 $\bar{\boldsymbol{W}}$ 顺序无重叠地分为 k 组：

$$\tilde{\boldsymbol{W}}=(\tilde{\boldsymbol{W}}_1,\tilde{\boldsymbol{W}}_2,\cdots,\tilde{\boldsymbol{W}}_k),\quad \bar{\boldsymbol{W}}=(\bar{\boldsymbol{W}}_1,\bar{\boldsymbol{W}}_2,\cdots,\bar{\boldsymbol{W}}_k)\tag{6}$$

则每组包含 $m/k=n\times\lceil r/l\rceil$ bit的水印信息，即 $|\tilde{\boldsymbol{W}}_i|=|\bar{\boldsymbol{W}}_i|=n\times\lceil r/l\rceil$；

② 将每个 $\tilde{\boldsymbol{W}}_i$ 和 $\bar{\boldsymbol{W}}_i$ 无重叠地分为 $\lceil r/l\rceil$ 组：

$$\tilde{\boldsymbol{W}}_i=(\tilde{w}_{i,1},\tilde{w}_{i,2},\cdots,\tilde{w}_{i,k}),\quad \bar{\boldsymbol{W}}_i=(\bar{w}_{i,1},\bar{w}_{i,2},\cdots,\bar{w}_{i,k})\tag{7}$$

则每组包含 n bit的水印信息，即 $|\tilde{w}_{i,j}|=|\bar{w}_{i,j}|=n$；

③ 设定检测向量 $\boldsymbol{D}_i=(d_{i,1},d_{i,2},\cdots,d_{i,r})$，初始设置 $d_{i,j}=0$，表示 $\tilde{\boldsymbol{X}}$ 中的第 j 个图像块通过认证，将其按顺序无重叠的划分为大小为 l 的组：

$$\boldsymbol{D}_i=(D_{i,1},D_{i,2},\cdots,D_{i,k})\tag{8}$$

则 $D_{i,j}$ 对应于水印信息 $\tilde{w}_{i,j}$ 和 $\bar{w}_{i,j}$；

④ 对于 $D_{i,j}=(d_{i,(l-1)j+1},d_{i,(l-1)j+2},\cdots,d_{i,l\times j})$，令十进制数 $q=\tilde{w}_{i,j}\oplus\bar{w}_{i,j}$，若 $q\neq0$，则将该分组中的第 q 个图像块判断为不能通过认证，即设置 $d_{i,(l-1)j+q}=1$。

⑤ 对每个分组进行上述判断，则可确定检测向量 $\boldsymbol{D}_i=(d_{i,1},d_{i,2},\cdots,d_{i,r})$ 的值；

⑥ 对每个 $\tilde{\boldsymbol{W}}_i$ 和 $\bar{\boldsymbol{W}}_i$ 分别确定其 $\boldsymbol{D}_i$，则设定最终检测向量 $\boldsymbol{D}=(d_1,d_2,\cdots,d_r)$，其中 $d_j=\mathop{\mathrm{or}}\limits_{i=1}^{k}d_{i,j}$；

⑦ 根据 $\boldsymbol{D}=(d_1,d_2,\cdots,d_r)$ 判断图像块是否被篡改，即若 $d_j=1$，则说明 $\tilde{\boldsymbol{X}}$ 中的第 j 个图像块不能通过认证，否则该块通过认证；

⑧ 设置 $\boldsymbol{U}=(u_1,u_2,\cdots,u_r)$ 表示 k 重水印中将各块认证为不通过的个数，其中 u_j 由下式得到：

$$u_j=\sum_{i=1}^{k}d_{i,j}\tag{9}$$

可知 $0\leqslant u_j\leqslant k$；

⑨ 设置阈值 $1\leqslant u\leqslant k$，则对于每一个图像块 $\tilde{x}_j$，若 $u_j\geqslant u$，则将该图像块判定位篡改图像块，否则判定为非篡改图像块。

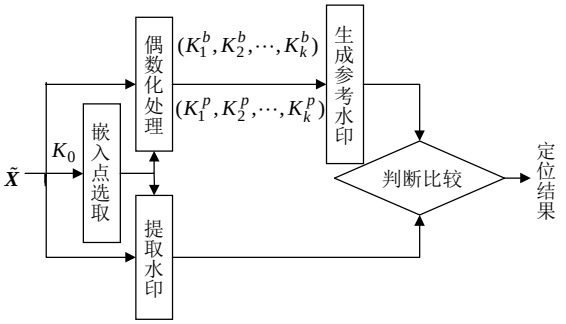


图2 篡改定位流程图

3 算法分析

3.1 篡改定位能力分析

本节分析在嵌入一重水印 $\boldsymbol{W}_i$ 情况下的篡改定位能力。设提取的水印为 $\tilde{\boldsymbol{W}}_i$，生成的参考水印为 $\bar{\boldsymbol{W}}_i$，其分别被分为 $\lceil r/l\rceil$ 组，对于其中的一个分组 $\tilde{\boldsymbol{W}}_i^j$ 和 $\bar{\boldsymbol{W}}_i^j$ 来说，与其对应的图像块的认证向量为 $\boldsymbol{D}_i^j=(d_i^{(l-1)j+1},d_i^{(l-1)j+2},\cdots,d_i^{l\times j})$。分如下4种情况分析本文算法的篡改能力：

1) 在 $\boldsymbol{D}_i^j$ 对应的图像块未被修改，且提取的水

印信息 $\tilde{\boldsymbol{W}}_i^j$ 未发生改变的情况下，有 $\tilde{\boldsymbol{W}}_i^j=\bar{\boldsymbol{W}}_i^j$，因此由 $\tilde{\boldsymbol{W}}_i^j$ 和 $\bar{\boldsymbol{W}}_i^j$ 按位进行异或得到的十进制数 $v=0$，表明 $\boldsymbol{D}_i^j$ 对应的所有图像块均未被篡改，通过认证。

2) 在 $\boldsymbol{D}_i^j$ 对应的图像块中只有一个被篡改，且提取的水印信息 $\tilde{\boldsymbol{W}}_i^j$ 未发生改变的情况下，设 $d_i^{(l-1)j+t}$ 被篡改，则生成参考水印时，得到的该块的Hash值肯定产生变化，而对该块的Hash值进行逐位异或后产生的bit位 b_t 将有1/2的概率发生变化。在 b_t 发生变化的情况下，由矩阵编码过程可知，生成的 n bit水印信息 $\bar{\boldsymbol{W}}_i^j=(w_1,w_2,\cdots,w_n)$ 中，将会在十进制数 $t=t_1t_2\cdots t_n$ 的比特位为1的对应位置上发生变化。因此，在提取的参考水印 $\tilde{\boldsymbol{W}}_i^j$ 未发生变化的情况下，由 $\tilde{\boldsymbol{W}}_i^j$ 和 $\bar{\boldsymbol{W}}_i^j$ 按位进行异或得到的十进制数 $v=t$，可确定第 t 个图像块 $d_i^{(l-1)j+t}$ 发生篡改。即在该情况下将有1/2的概率能准确检测出被篡改的图像块。

3) 在 $\boldsymbol{D}_i^j$ 对应的图像块中有多个块被篡改时，根据情况2)，由 $\tilde{\boldsymbol{W}}_i^j$ 和 $\bar{\boldsymbol{W}}_i^j$ 按位进行异或得到十进制数 $v=t$，只能定位到第 t 个图像块被篡改，因此将无法准确检测出所有的被篡改图像块，且会将一个未被篡改图像块检测为被篡改图像块，产生虚检。

4) 当提取的水印信息 $\tilde{\boldsymbol{W}}_i^j$ 发生变化的情况下，$\tilde{\boldsymbol{W}}_i^j$ 和 $\bar{\boldsymbol{W}}_i^j$ 按位进行异或得到 $v=t$，也将无法准确检测到被篡改图像块，且会产生一个虚检的图像块。

下面分析在含水印图像的被篡改率为 α 的情况下，本文算法的漏检率和虚检率。

3.2　漏检率分析

漏检率 P_{D} 是指将被篡改小块识别为篡改小块的概率。由以上分析可知，在一重水印情况下，对于一个被篡改小块 x_ℓ，及其所在的分组同时出现如下情况时，能够正确检测出该小块：

1) 对 x_ℓ 的Hash值进行异或后的bit值与原始bit值发生改变；2) 该小组中只包含该篡改小块，或者除 x_ℓ 外，还包含其他的被篡改的小块，且其他被篡改的小块的bit值与原始bit值均相同；3) 该小组所对应的水印信息未被篡改。

对于情况1)，由上节的分析，其发生的概率为：

$$P(A)=\frac{1}{2} \tag{10}$$

对于情况2)，由于对图像进行了置乱处理且每个图像块被篡改的概率为 α，则 x_ℓ 所处的分组中除 x_ℓ 外还有 $a\,(0\leqslant a\leqslant l-1)$ 个图像块被篡改的概率为：

$$P(B_1)=C_{l-1}^{a}\alpha^{a}(1-\alpha)^{l-1-a} \tag{11}$$

而对于这 a 个被篡改的图像块，其bit值均未发生变化的概率为：

$$P(B_2)=\left(\frac{1}{2}\right)^{a} \tag{12}$$

由于 B_1 和 B_2 相互独立，则情况2)发生的概率为：

$$P(B)=\sum_{a=0}^{l-1}P(B_1)P(B_2)=\sum_{a=0}^{l-1}C_{l-1}^{a}\alpha^{a}(1-\alpha)^{l-1-a}\left(\frac{1}{2}\right)^{a} \tag{13}$$

对于情况3)，由于整幅图像的篡改率为 α，因此所嵌入的每个水印bit的篡改率为 $\alpha/2$，对于一个分组来说，其水印信息共有 $n=\lceil\log_2(l+1)\rceil$ bit，因此这 n 个水印信息均未被篡改的概率为：

$$P(C)=\left(1-\frac{\alpha}{2}\right)^{n} \tag{14}$$

由于A、B、C相互独立，因此在一重水印的情况下，能够正确检测出该篡改图像块的概率是：

$$P_{\mathrm{D}}'=P(A)P(B)P(C) \tag{15}$$

而对于 k 重水印，由于各重水印独立进行认证，则在 k 重水印中仅有 $b<u$ 个水印能够正确检测出该篡改小块的概率为(即漏检的概率)：

$$P_{\mathrm{D}}=P\{b<u\}=\sum_{i=0}^{u-1}P\{b=i\}=\sum_{i=0}^{u-1}C_k^i(P_{\mathrm{D}}')^i(1-P_{\mathrm{D}}')^{k-i} \tag{16}$$

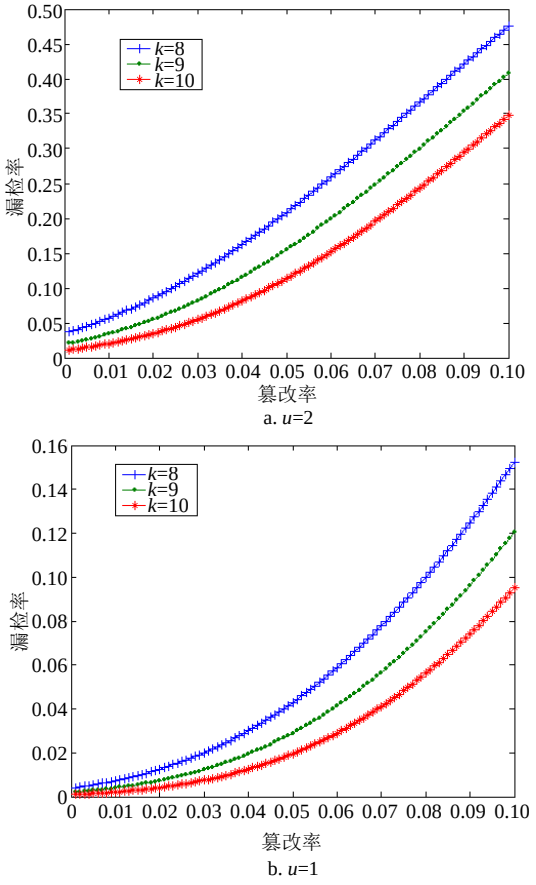


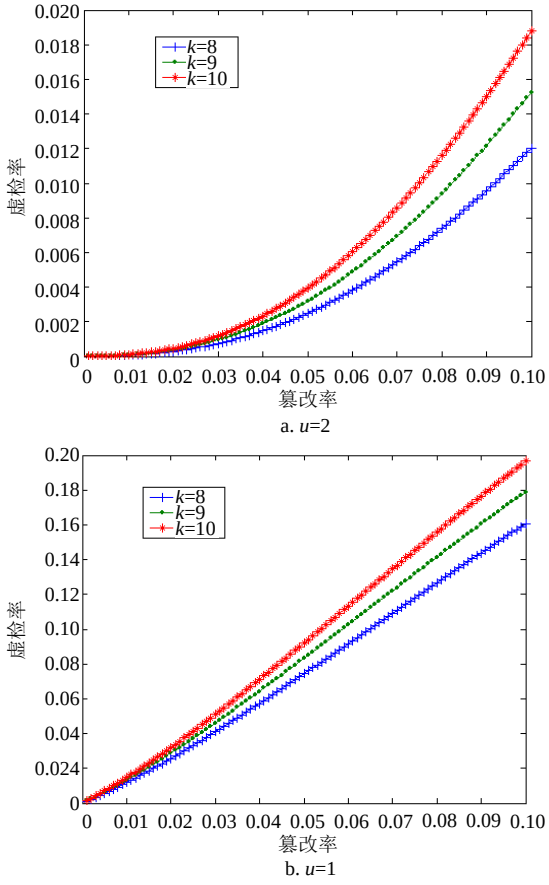
图3　漏检率随篡改率的变化情况

图3显示了在 $l=15$，$n=4$ 的情况下，k 分别取8、9、10时，漏检率随篡改率的变化情况。从图中可以看出，本文算法的漏检率随着嵌入水印的重数 k 的增大而减小，随阈值 u 的增大而增大。

3.3 虚检率分析

虚检率 P_{F} 是指将未被篡改小块检测为篡改小块的概率。在一重水印下，对于一个分组来说其出现以下两种情况之一，将会产生一个被虚检的小块：

1) 该小组的水印被篡改；2) 该小组中包含多个被篡改的小块，且至少有两个块的比特值发生变化。



a. u=2

b. u=1

图4 虚检率随篡改率的变化情况

对于情况1)，由上节的分析，其发生的概率为：

$$P(A)=1-\left(1-\frac{\alpha}{2}\right)^{n} \tag{17}$$

对情况2)，该小组中的任一图像块为篡改小块的概率为 α，而对于篡改小块来说，其bit值发生变化的概率为 $1/2$，则对于该分组中的任一图像块，其bit值发生变化的概率为 $\alpha/2$。则分组中至少有两个块的bit值发生变化的概率，即情况2)发生的概率为：

$$P(B)=1-P(\overline{B})=1-\left(1-\frac{\alpha}{2}\right)^{l}-l\frac{\alpha}{2}\left(1-\frac{\alpha}{2}\right)^{l-1} \tag{18}$$

则一重水印情况下，产生虚检块分组的概率为：

$$P(A\cup B)=P(A)+P(B)-P(AB)=1-\left(1-\frac{\alpha}{2}\right)^{n+l}-l\frac{\alpha}{2}\left(1-\frac{\alpha}{2}\right)^{n+l-1} \tag{19}$$

因此，对于 r/l 个分组来说，一重水印将产生 $P(A\cup B)r/l$ 个虚检的图像块，其虚检率为：

$$P_{\mathrm{F}}'=\frac{P(A\cup B)r/l}{r}=\frac{P(A\cup B)}{l} \tag{20}$$

对于 k 重水印，由于各重水印独立进行认证，则各水印产生的虚检块将独立随机分布在整幅图像中。则在 k 重水印中有 $c\geqslant u$ 个水印将同一个真实图像块判别为篡改图像块的概率(即虚检的概率)为：

$$P_{\mathrm{F}}=P\{c\geqslant u\}=1-P\{c<u\}=1-\sum_{i=0}^{u-1}P\{c=i\}=1-\sum_{i=0}^{u-1}C_k^i(P_{\mathrm{F}}')^i(1-P_{\mathrm{F}}')^{k-i} \tag{21}$$

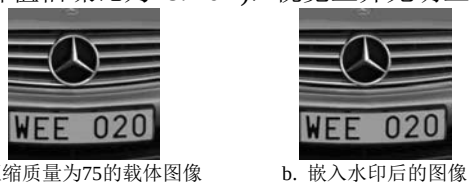
图4为 $l=15$，$n=4$ 的情况下，k 分别取8、9、10时，虚检率随篡改率的变化情况。从图中可以看出，本文算法的漏检率随着嵌入水印的重数 k 的增大而增大，同时，随着阈值 u 的增大而减小。

4 实验结果及分析

实验中所采用的图像块置乱为文献[10]提出的混沌置乱方法，所设置的分组大小 $l=15$，则每个分组生成 $n=4$ bit的水印信息，设置的水印重数 $k=8$，则平均每个分块所嵌入的水印信息长度约为：

$$x\approx\frac{n\times k}{l}=\frac{4\times 8}{15}\approx 2.13\ \ \text{bit} \tag{22}$$

图5给出了压缩质量为75的原始图像(大小为256×256，共包含1 024个图像块)，及嵌入水印后的图像(峰值信噪比为48.7 dB)，视觉上并无明显差异。



a. 压缩质量为75的载体图像　　b. 嵌入水印后的图像

图5 载体图像与含水印图像

a. 篡改率为0.4%的篡改图像　　b. u=2认证结果图像　　c. u=1认证结果图像

图6 篡改率为0.4%的篡改图像及其认证图像

图6a为篡改后的图像，将车牌号第3个字母“E”修改为“F”，篡改率为0.4%，即有4个图像块被篡改。按照第3节的分析，当阈值 $u=2$ 和 $u=1$ 时，漏检率

与虚检率分别约为0.043、8.7×10^{-6}和5.1×10^{-3}、4.5×10^{-3}。图6b为$u=2$时篡改检测的实际结果(黑色图像块为认证不通过图像块)，图6c为$u=1$时篡改检测的实际结果，可见能够完全检测出被篡改图像块，且无虚检的情况发生，与理论分析的结果完全相符。

图7a为将车牌号第3个字母“E”修改为“N”，篡改率为2.3%，即有24个图像块被篡改。按照第3节的分析，当阈值$u=2$时，其漏检率与虚检率分别约为0.095和4×10^{-5}，即约有2.3个篡改图像块被漏检，约有9×10^{-3}个未篡改图像块被虚检；当$u=1$时，其漏检率与虚检率分别约为1.4×10^{-2}和3.0×10^{-2}，即约有0.3个图像块被漏检，约有0.7个图像块被虚检。图7b、图7c分别为$u=2$和$u=1$时对其进行篡改检测的实际结果，可见实际结果与理论分析的结果基本相符。

a. 篡改率为2.3%的篡改图像　b. $u=2$ 认证结果图像　c. $u=1$ 认证结果图像

图7　篡改率为2.3%的篡改图像及其认证图像

a. 篡改率为4.6%的篡改图像　b. $u=2$ 认证结果图像　c. $u=1$ 认证结果图像

图8　篡改率为4.6%的篡改图像及其认证图像

图8a为将车牌号的第3个字母“E”修改为“N”，并将第一个数字“0”修改为“2”，篡改率为4.6%。按照第3节中的分析，分别取阈值为$u=2$和$u=1$时，其漏检率与虚检率分别约为0.189 41、2×10^{-3}和3.7×10^{-2}、6.7×10^{-2}，即分别约有9个图像块被漏检、0.098个图像块被虚检，1.8个图像块被漏检、3.2个图像块被虚检。图8b、图8c分别为$u=2$和$u=1$时篡改检测的实际结果，与理论分析的结果完全相符。

5　结　论

本文提出一种针对JPEG图像的块级认证算法，通过对块置乱后的图像进行分组，并使用矩阵编码生成水印信息，通过嵌入多重水印信息来共同定位篡改图像块。对算法的漏检率和虚检率进行了较详细的分析，并实验验证了理论分析结果的正确性。理论分析和试验仿真均表明，算法能够较准确定位到被篡改的图像块，实现JPEG图像的块级认证。

参　考　文　献

[1] 王国栋, 刘粉林, 汪萍, 等. 一种篡改检测与篡改定位分离的图像认证方案[J]. 计算机学报, 2007, 30(10): 1880-1888.
WANG Guo-dong, LIU Fen-lin, WANG Ping, et al. An image authentication scheme separating tamper detection from tamper location[J]. Chinese Journal of Computers, 2007, 30(10): 1880-1888.

[2] 王国栋, 刘粉林, 刘媛, 等. 一种能区分水印或内容篡改的脆弱水印算法[J]. 电子学报, 2008, 36(7): 1349-1354.
WANG Guo-dong, LIU Fen-lin, LIU Yuan, et al. An image authentication scheme with discrimination of tampers on watermark or image[J]. Acta Electronica Sinica, 2008, 36(7): 1349-1354.

[3] YANG Chun-wei, SHEN Jau-ji. Recover the tampered image based on VQ indexing[J]. Signal Processing, 2010(90): 331-343.

[4] HE Hong-jie, CHEN Fan, TAI Heng-ming, et al. Performance analysis of a block-neighborhood based self-recovery fragile watermarking scheme[J]. IEEE Transactions on Information Forensics And Security, 2012, 7(1): 185-196.

[5] 陈帆, 和红杰, 王宏霞. 用于图像认证的变容量恢复水印算法[J]. 计算机学报, 2012, 35(1): 154-162.
CHEN Fan, HE Hong-jie, WANG Hong-xia. Variable-payload self-recovery watermarking scheme for digital image authentication[J]. Chinese Journal of Computers, 2012, 35(1): 154-162.

[6] 金喜子, 姜文哲. 块级篡改定位的JPEG图像脆弱水印[J]. 电子学报, 2010, 38(7): 1585-1589.
JIN Xi-zi, JIANG Wen-zhe. Fragile watermarking capable of locating tampered of locating tampered blocks in JPEG images[J]. Acta Electronica Sinica, 2010, 38(7): 1585-1589.

[7] LIN C Y, CHANG S F. Semi-fragile watermarking for authenticating JPEG visual content[C]//Proceedings of the SPIE: Security and Watermarking of Multimedia Contents II. San Jose, USA: [s.n.], 2000, 3971: 140-151.

[8] LI Bao, TAO Xu. A new semi-fragile watermarking algorithm for image authentication[C]//Proceeding of World Congress on Intelligent Control and Automation. [S.l.]: [s.n.], 2008: 5928-5932.

[9] LI C T. Digital fragile watermarking scheme for authentication of JPEG images[J]. IEE Proceedings-Vision, Image, and Signal Processing, 2004, 151(6): 460-466.

[10] 高山青, 张士杰, 刘镔, 等. 一种基于四值混沌阵列的数字图像加密算法[J]. 中国图像图形学报, 2006, 11(2): 244-250.
GAO Shan-qing, ZHANG Shi-jie, LIU Bin, et al. An image encryption algorithm based on four value chaoticarray[J]. Chinese Journal of Image and Graphics, 2006, 11(2): 244-250.

编　辑　漆　蓉