第 56 卷 第 6 期 2007 年 6 月
1000-3290/2007/56(06)/3092-09

物 理 学 报
ACTA PHYSICA SINICA

Vol.56,No.6,June,2007

基于混沌的自嵌入安全水印算法*

和红杰† 张家树

(西南交通大学信号与信息处理四川省重点实验室,成都 610031)

(2006 年 6 月 27 日收到;2006 年 10 月 17 日收到修改稿)

利用混沌系统的伪随机性和初值敏感性,提出一种基于混沌的自嵌入安全水印算法.该算法以混沌初值为密钥生成混沌序列,根据混沌序列的索引有序序列随机生成图像块的水印嵌入位置.与现有的自嵌入算法相比,该算法实现了水印嵌入位置的随机选取,有效扩大了算法的密钥空间,且解决了自嵌入水印算法如何准确定位篡改块的问题.理论分析和仿真结果表明,该算法不仅提高了自嵌入水印算法的篡改定位的能力,而且进一步增强了算法抵抗向量量化攻击和同步伪造攻击的能力.

关键词:数字水印,混沌,脆弱水印,自嵌入

PACC:0545

1. 引 言

数字水印(Digital Watermarking)技术从提出至今不到十年的时间里,已成为信息科学前沿领域一个新颖且具有广泛应用前景的研究热点.根据水印的特性,数字水印可分为鲁棒水印(Robust Watermarking)[1—5]、半脆弱水印(Semi-Fragile Watermarking)[6]和脆弱水印(Fragile Watermarking)[7—13]三类.鲁棒水印主要用于版权保护,而脆弱水印和半脆弱水印主要用于鉴定多媒体数据的完整性和真实性.与传统的密码算法相同,数字水印算法应该是公开的,其安全性依赖于算法的密钥[13].

近年来,利用混沌系统的确定性和对初值的敏感性产生随机序列[14,15]和基于混沌构造各类密码算法[16,17]的研究,为安全水印算法研究开辟了新的途径.人们开始尝试利用混沌系统来提高数字水印算法的性能,主要有以下几种方法:(1)混沌序列作为水印信号[1—5];(2)实现水印的随机嵌入[6];(3)对水印信号的加密处理[7]等.目前,基于混沌的水印算法研究主要集中在鲁棒水印上,对半脆弱或脆弱水印算法的研究较少,而利用混沌来提高自嵌入水印算法安全性的研究还没见有报道.

自嵌入水印算法[10—12]是脆弱水印的一种,它除了满足不可见、篡改检测(脆弱性)、篡改定位和安全等脆弱水印的一般特性[18]之外,还能近似恢复被篡改的图像内容,充分展示了基于数字水印的多媒体(如数字图像)认证技术的优势.1999 年,Fridrich[10]首次提出了一种基于 DCT 的分块自嵌入脆弱水印算法,该算法将图像分为互不相交的 8×8 的图像块,对其高 7 位的 DCT 系数按一定的码长量化编码后,采用固定‘偏移值’嵌入另一图像块的最低位(简称 LSB),从而在定位图像篡改块的同时,还可以利用水印信息近似恢复被篡改图像块的内容.文献[11,12]指出‘偏移值’应由密钥控制以抵抗可能的伪造攻击[19,20],并分别给出了根据密钥生成图像块水印嵌入位置的方法.然而,他们生成水印嵌入位置的密钥空间较小,且图像块与相应水印嵌入位置成线性关系,因此不能抵抗文献[21]提出的同步伪造攻击,从而使自嵌入水印算法存在严重的安全隐患.

为提高自嵌入水印算法的安全性,本文提出一种基于混沌的自嵌入安全水印算法.该算法首先选择一个混沌映射,以混沌初值为密钥生成混沌序列,根据该混沌序列的索引有序序列随机生成图像块的水印嵌入位置.基于混沌的自嵌入水印算法实现了图像块水印的随机嵌入,不仅有效提高了算法的安全性,而且解决了自嵌入水印算法准确定位篡改块的难题.理论分析和实验仿真结果表明,该算法不仅

* 国家自然科学基金(批准号:60572027),四川省青年科技基金(批准号:03ZQ026-033),教育部新世纪优秀人才计划项目(批准号:NCET-05-0794),国防预研基金项目(批准号:51430804QT2201)和四川省应用基础研究项目(批准号:2006 J13-10)资助的课题.

† E-mail:hehojie@sohu.com

提高了自嵌入水印算法的篡改定位的能力，而且进一步增强了算法抵抗向量量化攻击和同步伪造攻击的能力.

2. 水印嵌入位置的安全性分析

在自嵌入水印算法中，水印嵌入偏移值[10,11]指定了图像块的水印的嵌入位置，第 i 个图像块的水印嵌入位置 $f(i)$ 与该图像块偏移值的关系为

$$f(i) = i + p_i,\ i = 1,2,\cdots,N, \qquad (1)$$

其中，N 表示图像块数，p_i 为第 i 个图像块的偏移值.生成偏移值的目的是为了确定图像块的水印的嵌入位置，因此在自嵌入水印算法中，偏移值和水印嵌入位置有相同的作用和性质.

自嵌入水印算法中，图像块 i 的水印信息嵌入在图像块 $f(i)$ 的最低位有两个好处：1）增加图像块之间的相关性，提高算法抵抗向量量化攻击[19]或拼贴攻击[20]的能力；2）降低图像块内容和相应水印信息同时被篡改的可能性，提高自嵌入算法的篡改恢复质量.不过，这也给算法的篡改定位带来了困难.因为一个图像块被篡改，会导致该图像块和另一个图像块（它的水印信息嵌入在被篡改图像块的最低位）同时被检测，如何判断他们中哪一个块被篡改，是自嵌入水印算法还没有完全解决的一个难题.同时，现有基于密钥生成水印嵌入位置的密钥空间较小，还不能满足自嵌入算法安全性的需要[21].下面结合自嵌入水印算法的特点并借鉴文献[11,12,21]对偏移值的分析，讨论自嵌入安全水印算法中水印嵌入位置 $f(i)$ 应满足的条件.

1）必要条件　根据水印嵌入位置的实际意义，$f(i)$ 须满足两个必要条件[21]：

·对 $\forall\ i \in [1,N]$，有 $f(i) \in [1,N]$，即每一个图像块水印都必须放在该图像中某个图像块的最低位；

·对 $\forall\ i,j \in [1,N]$，$i \neq j$，有 $f(i) \neq f(j)$，即两个图像块的水印不能放在同一个图像块的最低位.

2）篡改定位的要求　$f(i)$ 在整个图像中随机分布.

自嵌入水印算法中，一个图像块 X_i 的最低位嵌入的是另一个图像块 X_j（$j \neq i$）的水印信息，因此当 X_i 被篡改时，会导致它和 X_j 同时被检测.如何识别 X_i 和 X_j 这两种不同性质的被检测图像块，是解决自嵌入水印算法篡改定位的关键.对单个图像块的篡改，算法很难区分这两种不同类型的被检测图像块.但是，对图像块个数大于 1 的区域篡改，算法则可以根据其不同分布来实现篡改定位.

设被篡改的多个相邻图像块记为区域 A，相应的，水印信息被嵌入在区域 A 中的图像块记为集合 B（这些图像块不一定相邻）.当区域 A 被篡改时，A 和 B 中的图像块将同时被认证算法检测.如果对含水印图像中的任一区域 A，相应集合 B 中的图像块在整个图像中随机分布，如图 1（a）所示（区域 A 包含 4 个相邻的图像块的情况）.此时尽管 A，B 同时被检测，但区域 A 和集合 B 中的图像块的分布明显不同.可以根据其周围被篡改图像块数来定位篡改（详见 3.2 节）.反之，如果 B 中的图像块也相邻（如图 1（b））,则算法定位篡改的难度和误判的可能性都会随之增大.由 $f(i)$ 和 i 的对应关系可知，含水印图像中任一区域 A 对应的集合 B 中的图像块在整个图像随机分布的充要条件是每个图像块的水印嵌入位置 $f(i)$ 在整个图像随机分布.

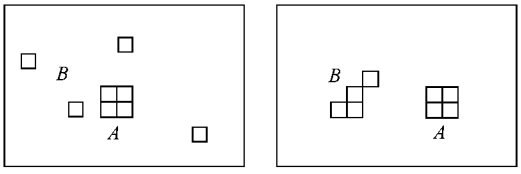

图 1　区域 A 最低位对应图像块的不同分布

3）篡改恢复的要求　图像块与其水印信息嵌入块之间的距离不能太近[10,11]；

在自嵌入水印算法中，利用相应水印中保存的信息能够近似恢复被篡改图像块的内容，然而，如果图像块和相应水印同时被篡改，则算法无法进行有效的篡改恢复.为降低图像块及其水印信息同时被破坏的可能性，$f(i)$ 与 i 之间的距离不能太近[10,11].

4）抵抗向量量化攻击的要求　不能形成多个固定的小认证链.

水印嵌入偏移值可以增加图像块之间的相关性，使自嵌入算法能够抵抗针对‘块独立（Block-wise independent）’脆弱水印算法的向量量化攻击[19]或拼贴攻击[20].但是，如果含水印图像形成多个固定的小认证链，攻击者仍然能以认证链为单位对含水印图像实施向量量化攻击或拼贴攻击.例如采用 $N/3$ 作为水印嵌入偏移值（N 为图像块数），当 N 能被 3 整除时，图像块及其相应的水印嵌入位置关系如图2

所示,即对 $\forall i \in [1, N/3]$, $\{y_i, y_{i+N/3}, y_{i+2N/3}\}$ 构成一个认证链.此时的自嵌入算法可以看作是以“认证链”为单位的“块独立”水印算法,因此,不能抵抗以认证链为单位的向量量化攻击或拼贴攻击.

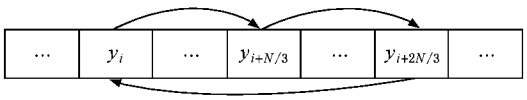

图 2 图像块及相应水印嵌入位置关系图

5) 抵抗同步伪造攻击的要求 偏移值或水印嵌入位置 $f(i)$ 由用户密钥 k 控制,密钥空间应足够大.

文献[21]指出,如果攻击者得到一幅或多幅利用相同密钥生成的水印图像并估计出水印嵌入偏移值,不需要知道水印加密密钥,通过同步伪造攻击可以在任意一幅不含水印的图像中伪造水印.因此偏移值应该由密钥控制且其密钥空间足够大,以提高自嵌入水印算法抵抗同步伪造攻击的能力.

3. 基于混沌的自嵌入水印算法

为保证自嵌入水印算法的优良特性及安全性,基于密钥生成水印嵌入位置不仅要有大的密钥空间,而且应满足篡改定位、篡改恢复等要求.本文基于混沌系统对初值的极端敏感性和良好的随机性等特点,提出一种基于混沌的自嵌入安全水印算法.该算法基于混沌生成的水印嵌入位置不仅可以满足第 2 节讨论的 5 个条件,而且解决了自嵌入水印算法如何定位篡改的难题.

3.1. 水印嵌入算法

基于混沌的自嵌入安全水印算法包括以下步骤:

Step1 先将原始图像 X 最低位置零,再将其分为互不相交的 8×8 的图像块 $\tilde{X}_i$, $i = 1,2,\cdots,N$ ($N = m \times n$ 为图像块数,即原始图像的大小为 $8m \times 8n$);

Step2 对图像块 $\tilde{X}_i$ 进行 DCT 变换,对其系数按一定的码长量化编码[11],生成基于图像块内容的二值编码块 $M_i = F(\tilde{X}_i)$,利用密钥 k_e 对其加密生成待嵌入水印块 W_i,公式描述为

$$W_i = E_{k_e}(M_i) = E_{k_e}(F(\tilde{X}_i)), \tag{2}$$

其中,$F(\cdot)$ 表示 DCT 变换及量化编码过程,$E(\cdot)$ 为加密函数.

Step3 基于混沌生成图像块的水印嵌入位置 $f(i)$, $i = 1,2,\cdots,N$,其方法如下:

(1) 根据给定的混沌初值 k_p (水印嵌入位置密钥),利用混沌映射生成长度为 N 的实值混沌序列 $S = (s_1, s_2, \cdots, s_N)$:

$$S = H(k_p, N), \tag{3}$$

其中,H 表示混沌映射,本文采用文献[7]中的混沌映射:

$$s_{n+1} = (1 + 0.3(s_{n-1} - 1.08) + 379s_n^2 + 1001 * z_n^2) \bmod 3, \tag{4}$$

其中,z_n 是任意的混沌序列(本文取 logistic 混沌映射).该混沌系统初值在 $(-1.5, 1.5)$ 之间时系统具有混沌吸引子,两个初值 s_1 和 s_2 分别记为 $k_{(1)}$ 和 $k_{(2)}$. logistic 混沌映射初值 z_1 在 $(0,1)$ 之间,该初值记为 $k_{(3)}$,则生成水印嵌入位置的密钥 $k_p = \{k_{(1)}, k_{(2)}, k_{(3)}\}$.

(2) 采用稳定排序法生成 S 的索引有序序列 A;

对混沌序列 $S = (s_1, s_2, \cdots, s_N)$ 采用稳定排序法生成有序序列 $S_A = (s_{a_1}, s_{a_2}, \cdots, s_{a_N})$,即 S_A 中的元素满足 $s_{a_1} \leqslant s_{a_2} \leqslant \cdots \leqslant s_{a_N}$,则索引有序序列 $A = (a_1, a_2, \cdots, a_i, \cdots, a_N)$;

(3) 令水印嵌入位置 $f(i) = a_i$, $i = 1,2,\cdots,N$.

为描述方便,将基于密钥 k_p 生成的水印嵌入位置分别记为

$$\boldsymbol{f}_{k_p} = F_A(H(k_p, N)), \tag{5}$$

其中,$\boldsymbol{f}_{k_p}$ 为基于密钥 k_p 生成的水印嵌入位置向量,F_A 为生成索引有序序列的过程.

Step4 根据水印嵌入位置向量 $\boldsymbol{f}_{k_p}$,将待嵌入水印块 W_i 嵌入图像块 $X_{f(i)}$ 的最低位,生成含水印图像块

$$Y_{f(i)} = \tilde{X}_{f(i)} + W_i. \tag{6}$$

3.2. 篡改检测与恢复算法

本文算法在定位篡改时需考察图像块周围被篡改的情况,用 $N_{-8}(i)$ 表示图像块 i 的 8 邻域,按从上到下、从左到右的顺序编号时,图像块 i 的 8 邻域包括图像块的编号如图 3 所示(图像的边界处补 0),即有

$$N_{-8}(i) = \{i \pm 1, i \pm n, i + n \pm 1, i - n \pm 1\}, \tag{7}$$

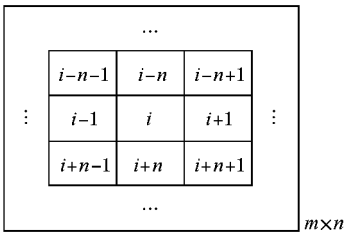

图 3　图像块 i 的 8 邻域

篡改检测和恢复算法步骤如下：

Step1　采用与水印嵌入相同的方法，将被测图像 Y^* 分为 8×8 的图像块 Y_i^*（$i=1,2,\cdots,N$）；

Step2　取被测图像块 Y_i^* 的最低位，利用解密密钥 k_d 对其解密，得到 Y_i^* 最低位恢复的二值编码块

$$Ml_i^* = D_{k_d}(LSB(Y_i^*)), \tag{8}$$

其中 $LSB(\cdot)$ 表示取图像块的最低位，k_d 为解密密钥.

Step3　将图像块 Y_i^* 的最低位置零，按水印嵌入算法 Step2 生成图像块内容 $\tilde{Y}_i^*$ 的二值编码块

$$M_i^* = F(\tilde{Y}_i^*). \tag{9}$$

Step4　根据密钥 k_p，按相同方法生成水印嵌入位置向量 $\boldsymbol{f}_{k_p} = \{f(1),\cdots,f(i),\cdots,f(N)\}$；

Step5　对任一图像块 Y_i^*，设 T_i 表示 $N_{\_8}(i)$ 中满足 $M_j^* \neq Ml_{f(j)}^*$（$j\in N_{\_8}(i)$）的图像块的个数，则有

1）如果 $M_i^* = Ml_{f(i)}^*$，判定 Y_i^* 没有被篡改；

2）如果 $M_i^* \neq Ml_{f(i)}^*$ 且 $T_i = T_{f(i)} = 0$，判定 Y_i^* 没有被篡改（孤立篡改块不给出定位信息），并设置标志位 F 记录执行该操作的次数；

3）如果 $M_i^* \neq Ml_{f(i)}^*$ 且 $T_i \geqslant T_{f(i)}$（$T_i = T_{f(i)} = 0$ 除外），则判定 Y_i^* 被篡改，否则判定 Y_i^* 没有被篡改.

Step6　如果标志位 $F>0$，给出图像可能受到随机篡改或认证设备攻击[21]提示信息；

Step7　若检测出 Y_i^* 块被篡改，则用图像块 $Y_{f(i)}^*$ 中的水印经过解码、逆量化和逆 DCT 变换可近似恢复被篡改的块 Y_i^*[11].

上述篡改检测与恢复算法中，Step5 描述了自嵌入水印算法定位篡改的方法.需要注意的是，该算法对单个篡改块不给出图像块被篡改的位置（Step5（2）），仅提示被测图像受到随机篡改（Step6）.不过，这个“缺点”是可以接受的.由于单个图像块篡改对图像的真实性影响不大，而且可以避免攻击者通过认证设备攻击得到图像块水印的嵌入位置，进而对少数图像块实施同步伪造攻击.

4. 算法性能分析及实验仿真

本节采用理论分析与实验仿真相结合，通过讨论(5)式生成的水印嵌入位置 $f(i)$ 是否满足第 2 节的 5 个条件，来分析本文自嵌入水印算法的性能，最后给出算法对区域篡改的实验仿真结果.

4.1. $f(i)$ 的必要条件

根据(5)式可知，基于密钥 k_p 得到的水印嵌入位置等于混沌序列的索引有序序列，即 $f(i)=a_i$.下面证明 $f(i)$ 满足前面第 2 节提出的必要条件.

·基于密钥生成的混沌序列 $S=(s_1,s_2,\cdots,s_N)$，元素 s_i 下标 i 为元素的地址，$a_i\in[1,N]$.因此按上述算法生成的水印嵌入位置 $f(i)$ 满足：$\forall\, i\in[1,N]$，$f(i)\in[1,N]$.

·对于序列 $S=(s_1,s_2,\cdots,s_N)$，排序前后序列的元素个数相同且任何一个元素 s_i 在有序序列中有且仅有一个位置，因此 $\forall\, i,j\in[1,N]$，$i\neq j$，s_i 和 s_j 在有序序列中的位置 $a_i\neq a_j$，即 $f(i)\neq f(j)$.

4.2. 篡改定位的要求

考察 $f(i)$ 是否在整个图像随机分布，即验证 $f(i)$ 的取值是否在整数区间 $[1,N]$ 上均匀分布.下面利用统计分析法来验证 $f(i)$ 在区间 $[1,N]$ 上的分布特性，具体方法如下：

1）将整数区间 $[1,N]$ 分为大小相同的小区间 $[aq+1,(a+1)q]$，其中 $a=0,1,\cdots,(N/q)-1$，q 为区间长度（N/q 为整数）；

2）利用随机数发生器产生 M 个密钥 $k_p(j)$ $j=1,2,\cdots,M$，按(5)式产生水印嵌入位置向量 $\boldsymbol{f}_{k_p(j)}$；

3）对给定的 i，统计 M 个向量 $\boldsymbol{f}_{k_p(j)}$ 中 $f(i)$ 落入区间 $[aq+1,(a+1)q]$（$a=0,1,\cdots,(N/q)-1$）的次数（Counts）.

当 $M=5000$，$N=1000$，$q=10$ 时，区间的个数为 $N/q=100$，$f(i)$ 落入每个区间的平均次数为 $Mq/N=50$.在上述条件下，i 分别等于 11，100，500，850 时，

$f(i)$的统计分布如图 4 所示.由图 4 可以看出,不同 i 的$f(i)$的统计分布相似,且其值在均值 50 附近变化.因此,可以认为基于混沌生成的水印嵌入位置 $f(i)$在区间$[1,N]$上近似均匀分布.能够满足自嵌入水印算法篡改定位的要求,4.6 的实验仿真结果也验证这一点.

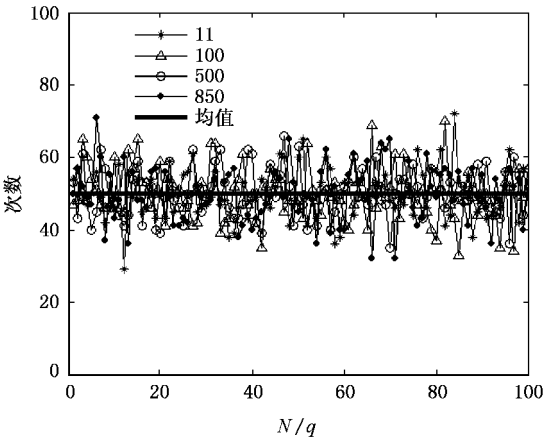

图 4 部分 $f(i)$的统计分布图

4.3. 篡改恢复的要求

为验证基于混沌随机生成水印嵌入位置 $f(i)$与图像块 i 的位置关系,定义 $f(i)$与 i 之间的距离为

$$D(i) = |f(i) - i|. \tag{10}$$

根据 $f(i)$与 i 的实际意义,$D(i)$为整数且取值范围为$D(i) \in [0, \max(i-1, N-i)]$,其中,$\max(\cdot)$表示取两个数中的最大值.由 4.2 的分析可知,$f(i)$在整数区间$[1,N]$上近似服从均匀分布,下面讨论随机变量 $D(i)$的数学期望 $E(D(i))$.

因为 $D(i) \in [0, \max(i-1, N-i)]$,所以 $D(i-1)$和 $D(N-i)$具有相同的取值范围和概率分布,即随机变量 $D(i)$的数学期望 $E(D(i))$关于 $(N+1)/2$ 对称.

当 $0 < i \leqslant [N/2]$时,

$$D(i) \in \begin{cases} 0, f(i) = i, \\ [1, i-1], f(i) \in [1, i-1], \\ [1, i-1], f(i) \in [i+1, 2i-1], \\ [i, N-i], f(i) \in [2i, N], \end{cases} \tag{11}$$

所以当 $0 < i \leqslant [N/2]$时,$D(i)$的概率分布为

$$P(D(i)) = \begin{cases} 1/N, D(i) = 0, \\ 2/N, D(i) \in [1, i-1], \\ 1/N, D(i) \in [i, N-i], \end{cases} \tag{12}$$

故 $D(i)$的数学期望 $E(D(i))$为

$$\begin{aligned} E(D(i)) &= \sum_{D(i)=0}^{N-i} D(i) P(D(i)) \\ &= 0 + \sum_{D(i)=1}^{i-1} \frac{2}{N} D(i) + \sum_{D(i)=i}^{N-i} \frac{1}{N} D(i) \\ &= \frac{2}{N} \frac{i(i-1)}{2} + \frac{1}{N} \frac{N(N-2i+1)}{2} \\ &= \frac{i(i-1)}{N} + \frac{N-2i+1}{2} \\ &= \frac{1}{N} i^2 - \left(1 - \frac{1}{N}\right) i + \frac{N+1}{2} \end{aligned} \tag{13}$$

所以,基于混沌随机生成水印嵌入位置 $f(i)$ 与图像块 i 之间的距离的数学期望为

$$E(D(i)) = \begin{cases} i^2/N - (1 - 1/N) i + (N+1)/2, 1 \leqslant i \leqslant [N/2], \\ E(D(N - i + 1)), [N/2] < i \leqslant N. \end{cases} \tag{14}$$

由(14)式可知,当 $i = 1$ 或 N 时,$E(D(i))$取得最大值 $\max(E(D(i))) = (N-1)/2$;当 $i = [(N+1)/2]$时,$E(D(i))$取得最小值 $\min(E(D(i))) \approx N/4$,说明基于混沌随机选取偏移值可以满足 i 与 $f(i)$之间的距离要求.

为验证基于混沌生成的水印嵌入位置满足 i 与 $f(i)$之间的距离要求及上述理论分析的正确性,分别取 N 等于 200 和 1024 并随机选取 1000 个密钥 k_p 进行测试,测试结果如图 5(a)和(b)所示.图中实线是按(14)式求出的 $E(D(i))$的值,"○"为随机选取 1000 个密钥计算得到的 $D(i)$的平均值.显然实验仿真结果和理论分析相一致,说明基于混沌映射随机生成水印嵌入位置能够满足自嵌入水印算法篡改恢复的要求.

4.4. 抵抗向量量化攻击

文献[11]指出,为提高自嵌入水印算法抵抗向量量化攻击的能力,应使所有图像块构成一个认证

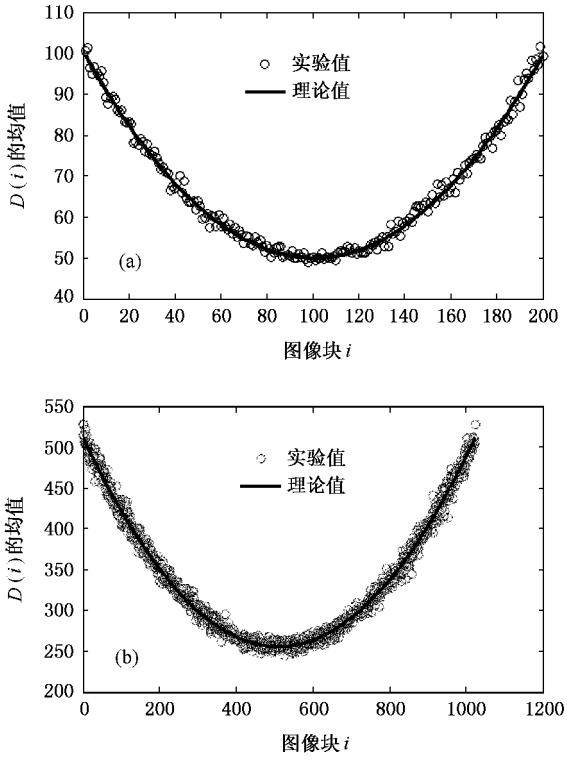

图 5　$E(D(i))$的实验与理论值比较　(a) $N=200$；(b) $N=1024$

链，此时要求偏移值 p 与图像块数 N 互素，即 p 与 N 的最大公约数为 1. 而满足该条件会大大缩小偏移值的取值范围.

基于混沌系统随机生成偏移值可扩大偏移值的取值范围，同时不可避免的会产生多个认证链. 但基于混沌随机选取偏移值构成认证链的个数不多，且认证链的个数及每个认证链包含的图像块数是随机的，下面通过实验仿真来验证：

1) 认证链个数是随机的：图 6 是随机选取 200 个密钥，当 N 等于 100，1000 和 5000 时认证链个数. 由图示可以看出，认证链的个数在一定范围内近似随机分布，认证链的个数随图像块的增多而增大，但是增大的速度很慢.

2) 认证链个数与图像块数 N 的关系：对区间[10，5000]中的每个整数 N，随机选取 300 个密钥计算认证链个数的平均值，图 7 为 N 与认证链个数的关系图，从图中可以看出，认证链个数随 N 的增大而增大，但增大的速度缓慢，当 $N=5000$ 时，认证链个数的均值仍然小于 10. 因此，基于混沌随机选取偏移值形成的认证链个数不会太多.

3) 认证链包含的图像块是随机的：为验证认证链及每个认证链中的图像块数与密钥的关系，随机

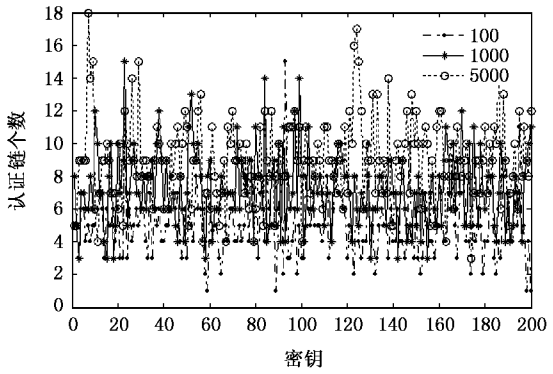

图 6　密钥-认证链个数的关系图

选取 200 个密钥，计算 N 的认证链个数及每个认证链中包含的图像块数. 图 8 示出 N 等于 100，500 和 1000 时，包含第一个图像块的认证链中所包含图像块的个数，由图示可知，该认证链包含的图像块数是随机的，且在(1，N)中近似随机分布. 因此在攻击者不知道密钥的情况下，很难确定哪些图像块构成一个认证链.

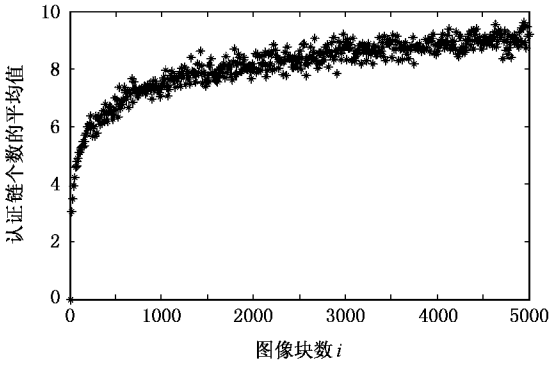

图 7　N-认证链个数的关系图

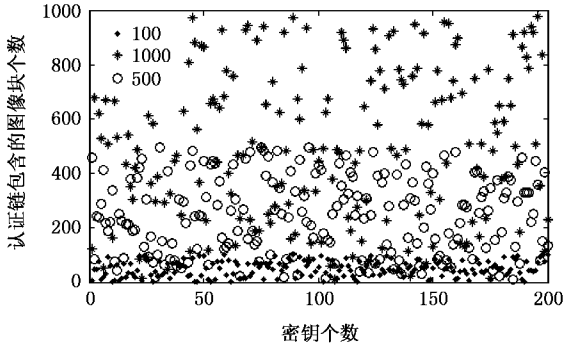

图 8　包含第一个图像块的认证链中的图像块数

根据前述的实验结果可知，基于混沌随机生成

偏移值形成认证链的个数不多，且认证链的个数及每个认证链包含的图像块数都是随机的，攻击者不知道密钥很难确定哪些图像块构成一个认证链，因此，基于混沌随机生成偏移值的自嵌入水印算法能够有效抵抗向量量化攻击.

4.5. 抵抗同步伪造攻击

基于混沌映射生成偏移值，密钥空间仅依赖于混沌系统对初值的敏感性，与图像的大小(图像块 N)无关.

本文采用混沌映射生成水印嵌入位置的密钥 $k_p = \{k_{(1)}, k_{(2)}, k_{(3)}\}$，其三个子密钥 $k_{(1)}$，$k_{(2)}$ 和 $k_{(3)}$ 分别为不同区间上的实数.由实数的性质可知任意区间内的实数有无穷多个，然而由于计算机的位长效应，当 k_p 的变化小于 δ，即 $k \in O(k_i, \delta)$(如图 9 所示) 时，有 $H(k, N) = H(k_i, N)$.图 9 中任意 k_i 的 δ 邻域 $O(k_i, \delta)$，$i = 2, 3, \cdots, n-1$(k_1, k_n 为半邻域) 为产生混沌序列相同点的集合，该集合可看作密钥的一个值，因此 k 的密钥空间为实数区间 (k_1, k_n) 内互不重叠的 δ 邻域的个数，即

$$I_k = \frac{k_n - k_1}{2\delta} + 1. \tag{15}$$

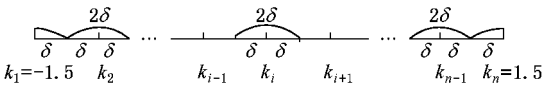

图 9 混沌密钥空间示意图

为求出邻域半径 δ，定义密钥 k 和 $k + \Delta k$ 生成的水印嵌入位置向量的平均距离为

$$M_{\Delta k}^{k} = \frac{1}{N}\sum_{i=1}^{N} | f_k(i) - f_{k+\Delta k}(i) |, \tag{16}$$

其中，k 为任取的密钥，Δk 为密钥 k 的变化量.由混沌对初值的敏感性可知，当密钥变化量 Δk 大于 δ 时，水印嵌入位置向量 f_k 和 $f_{k+\Delta k}$ 不同且为随机的；当密钥变化量 Δk 小于或等于 δ 时，$M_{\Delta k}^{k} = 0$.

为考察整个密钥空间的平均特性，从密钥空间 I_k 中随机选取密钥 $k^{(n)} = \{k_{(1)}^{(n)}, k_{(2)}^{(n)}, k_{(3)}^{(n)}\}$ $(n = 1, 2, \cdots, \Phi)$，定义密钥变化量的平均距离 $\overline{M}_{\Delta k}$：

$$\overline{M}_{\Delta k} = \frac{1}{\Phi}\sum_{n=1}^{N_1} M_{\Delta k}^{k^{(n)}}. \tag{17}$$

测试密钥 $k_{(1)}$ 的邻域半径 δ_1 取 $N = 1024$，在密钥空间中随机选取三组密钥，密钥个数分别为 100，300 和 1000.固定密钥 $k_2^{(n)}$ 和 $k_3^{(n)}$，仅对密钥 $k_{(1)}^{(n)}$ 变化，变化量 Δk 分别取 10^{-i}，$i = 1, 2, \cdots, 20$，根据(16)式分别计算出这三组密钥的 $\overline{M}_{\Delta k}$，绘制出当 $N = 1024$ 时 k_1 的 $\Delta k - \overline{M}_{\Delta k}$ 的三条曲线，如图 10 的上部所示，三条曲线分别用‘ * ’，‘ – ’和‘ + ’显示.图中横坐标为密钥 k_1 的变化量 Δk 的负对数，纵坐标为密钥变化量的平均距离 $\overline{M}_{\Delta k}$.由图 10 可以看出这三条曲线基本重合，说明本文采用的混沌系统具有良好的随机性.

为验证密钥空间与图像块数目 N 的关系，取 $N = 100$，重复上述试验过程绘制出当 $N = 100$ 时 $k_{(1)}$ 的 $\Delta k - \overline{M}_{\Delta k}$ 三条曲线，如图 10 下部所示.无论 N 等于 100 还是等于 1024，由图 10 的两条 $\Delta k - \overline{M}_{\Delta k}$ 曲线可以看出，当 $\Delta k < 10^{-12}$ 时，$\overline{M}_{\Delta k} \approx N/3$，可见算法的密钥空间与图像 N 的个数无关.密钥 k_1 的邻域半径 $\delta_1 < 10^{-12}$，由(15)式计算可得 k_1 的密钥空间 $I_{k_1} = \dfrac{1.5 + 1.5}{2\delta_1} > \dfrac{3}{2} \times 10^{12}$.

按相同的方法对密钥 $k_{(2)}$ 和 $k_{(3)}$ 测试，测试结果分别如图 11 和图 12 所示，按相同的计算方法可得 $k_{(2)}$ 和 $k_{(3)}$ 密钥空间分别为 $I_{k_2} > \dfrac{3}{2} \times 10^{15}$ 和 $I_{k_3} > 10^{16}$.因此该算法的密钥空间 $I_k = I_{k_1} \times I_{k_2} \times I_{k_3} > \dfrac{9}{4} \times 10^{43}$，且不受图像大小的影响.

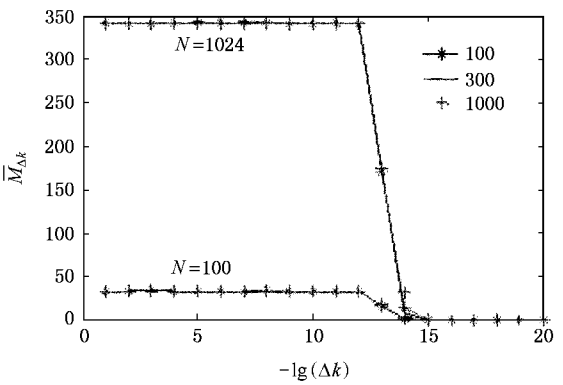

图 10 密钥分量 k_1 的 $\Delta k - \overline{M}_{\Delta k}$ 关系曲线

显然，基于混沌随机生成偏移值的密钥空间比文献[11,12]的密钥空间大得多，有效降低了攻击者利用穷举攻击估计出偏移值密钥的可能性.同时，由于算法对单个图像块篡改不给出位置信息，使得攻击者通过有目的访问认证设备也无法得到任何图像块的水印嵌入位置.因此，本文提出基于混沌的自嵌入水印算法能有效抵抗同步伪造攻击[21].

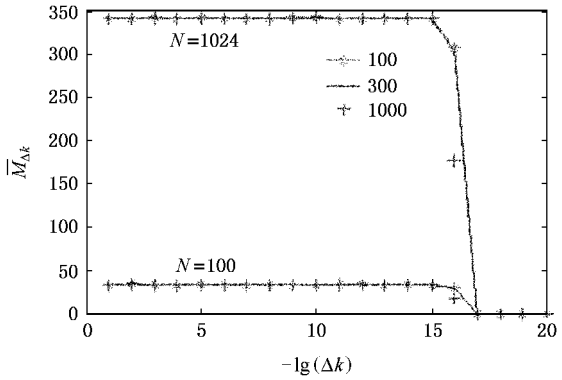

图 11　密钥分量 k_2 的 $\Delta k - \overline{M}_{\Delta k}$ 关系曲线

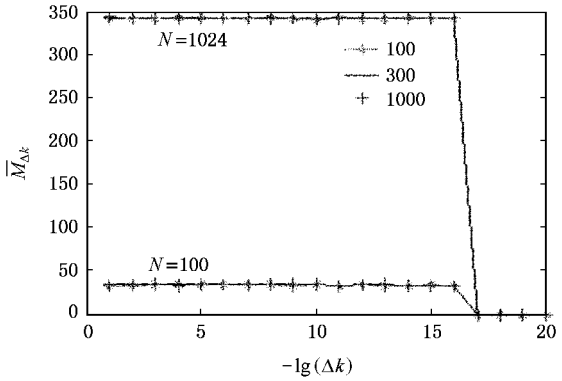

图 12　密钥分量 k_3 的 $\Delta k - \overline{M}_{\Delta k}$ 关系曲线

4.6. 仿真结果

为验证本文算法定位篡改和抵抗拼贴攻击的能力，对大小为 376 × 288 的“Napo Leon”和“MonaLisa”图像，利用本文提出的水印嵌入算法分别生成含水印图像，分别如图 13 (a)和(b)所示. 利用 Photoshop 编辑软件将“MonaLisa”的头部保持相对位置不变拼贴在“Napo Leon”的头部，得到的被篡图像如图13(c)所示. 图 13(d)为对(c)的篡改检测结果，篡改区域外被检测的篡改点是由于它们的水印信息改变造成的，即这些图像块的水印信息嵌入在篡改区域中某图像块的最低位. 利用本文提出的篡改定位算法，得到的篡改定位结果如图 13(e)所示，可以看出即使对这种恶意的“拼贴攻击”，算法也能准确定位图像被篡改的位置. 不过由于图像中被篡改的区域较大，因此篡改定位结果中存在少数误判的图像块.

上述理论分析和图 4—13 的仿真结果表明：基于混沌随机生成水印嵌入位置，不仅能满足自嵌入水印算法篡改定位、篡改恢复、抵抗向量量化攻击和抵抗同步伪造攻击等要求，而且还解决了自嵌入算法准确定位篡改的难题.

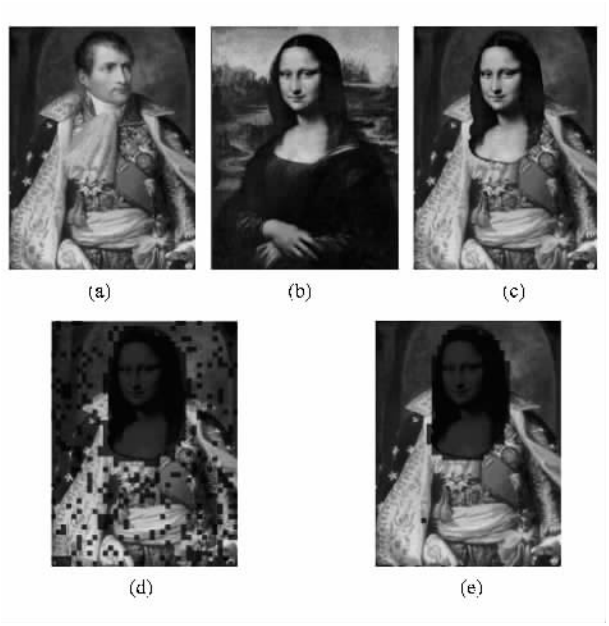

图 13　拼贴图像的篡改定位　(a)含水印的 Napo Leon；(b)含水印的 Monalisa；(c)对(a)的篡改图像；(d) 对(c)的篡改检测结果；(e) 对(c)的篡改定位结果

5. 结　论

本文结合混沌系统和自嵌入水印算法的特点，提出一种基于混沌的自嵌入安全水印算法. 该算法以混沌初值为密钥随机生成图像块的水印嵌入位置，有效扩大了算法的密钥空间. 理论分析和仿真结果表明，该算法不仅能够有效判定篡改块，而且提高了算法抵抗向量量化攻击和同步伪造攻击能力，大大提高了自嵌入水印算法的安全性.

[1] Zhao D W, Chen G R, Liu W B 2004 *Chaos, Solitons Fractals*, **22** 47

[2] Feng G R, Jiang L G, He C, Xue Y 2006 *Chaos Solitons & Fractals* **27** 580

[3] Tefas A, Nikolaidis A, Nikolaidis N, Solavhidis V, Tsekeridou S, Pitas I 2003 *Chaos Solitons Fractals* **17** 567
[4] Lu W, Lu H T, Chung F L 2005 *ICMLC* 18
[5] Chen S Y, Leung H 2005 *IEEE Trans. on Image. Proc.* **14** 1590
[6] Ding K, He C, Jiang L G, Wang H X 2005 *IEICE Trans. Fund.* E88-A(3) 787
[7] He H J, Zhang J S, Tai H M 2006 Proceedings of the 2006 International Conference on Computational Intelligence and Seculity, Guanzhou, China, Part2 1180
[8] Wong P W and Memon N 2001 *IEEE Trans. Image. Proc.* **10** 1593
[9] Celik M U, Sharma G, Tekalp A M, Saber E 2005 *IEEE Trans. Image. Proc.* **14** 253
[10] Fridrich J, Goljan M 1999 *ICIP '99*, Kobe, Japan, October 25 – 28 1999
[11] Zhang H B, Yang C 2004 *Acta Elec.* **32** 196 (in Chinese) [张鸿宾、杨 成 2004 电子学报 **32** 196]
[12] Lin P L, Nsieh C K, Huang P W 2005 *Pattern Recognition* **38** 2519
[13] Fridrich J 2002 *Proc. SPIE* 4675
[14] Yu J J, Cao H F, Xu H B, Xu Q 2006 *Acta Phys. Sin.* **55** 29 (in Chinese) [于津江、曹鹤飞、许海波、徐 权 2006 物理学报 **55** 29]
[15] Wang L, Wang F P, Wang Z J 2006 *Acta Phys. Sin.* **55** 3964 (in Chinese) [王 蕾、汪芙平、王赞基 2006 物理学报 **55** 3964]
[16] Wang X M, Zhang J S, Zhang W F 2005 *Acta Phys. Sin.* **54** 5566 (in Chinese) [王小敏、张家树、张文芳 2005 物理学报 **54** 5566]
[17] Guo X F, Zhang J S 2006 *Acta Phys. Sin.* **55** 4442 (in Chinese) [郭现峰、张家树 2006 物理学报 **55** 4442]
[18] Zhu B, Swanson M, Tewfik A 2004 *IEEE Sig. Proc. Mag.* **3** 40
[19] Holliman M, Memon N 2000 *IEEE Trans. Image. Proc.* **9** 432
[20] Fridrich J, Goljan M, Memon N 2002 *Electronic Imaging.* **11** 262
[21] He H J, Zhang J S, Wang H X 2006 *IJCSNSB* **6**(1B) 251

A chaos-based self-embedding secure watermarking algorithm *

He Hong-Jie† Zhang Jia-Shu
(*Sichuan Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu* 610031, *China*)
(Received 27 June 2006; revised manuscript received 17 October 2006)

Abstract

This paper presents a novel chaos-based self-embedding secure watermarking algorithm based on the chaotic pseudorandom ness and sensitivity to initial value of chaotic map. In the proposed approach, the real-valued chaotic sequence, which is generated by the given initial value as secret key, is used to produce the ordered indices as the watermark embedding positions. The proposed algorithm has a larger key space than the existing self-embedding watermarking algorithms due to its watermark embedding positions being randomly selected by the secret key. Furthermore, the problem, how to locate the tampered image block in the self-embedding watermarking algorithm has been solved successfully. Theoretical analysis and simulation results show that our scheme not only improves the abilities of tamper localization, but also has better security against many attacks, including the vector quantization and synchronous counterfeiting attacks.

Keywords: digital watermarking, chaos, fragile watermarking, self-embedding
PACC: 0545

* Project supported by the National Natural Science Foundation of China (Grant No. 60572027), by the Sichuan Youth Science & Technology Foundation of China (Grant No. 03ZQ026-033), by the Program for New Century Excellent Talents in Ministry of Education of China (Grant No. NCET-05-0794), by the National Defense Pre-research Foundation of China (Grant No. 51430804QT2201) and by the Application Basic Foundation of Sichuan Province, China (Grant No. 2006 J13-10).

† E-mail: hehojie@sohu.com