一种基于混沌 Hash 函数的脆弱水印算法

杨 超,何小海
YANG Chao,HE Xiao-hai

四川大学 电子信息学院 图像信息研究所,成都 610064
Image Information Institute,School of Electronics and Information Engineering,Sichuan University,Chengdu 610064,China
E-mail:yyccvipyyccvip@sina.com

YANG Chao,HE Xiao-hai.Novel fragile watermarking scheme based on chaotic hash function.Computer Engineering and Applications,2008,44(17):74-77.

Abstract: Aiming at the bug of current block based fragile watermarking scheme,a novel fragile watermarking scheme based on chaotic hash function is proposed in this paper.Firstly chaotic hash function is implemented on each image block to generate binary image,then XOR the binary image with the chaotic binary image generated by chaotic system to attain binary fragile watermark,after being scrambled,the binary fragile watermark is embedded into LSB plane of original image;difference image is used to realize authentication.Theoretical analysis and experimental results show the proposed scheme has good invisibility,and ability of resisting VQ attack,differentiating the tamper type,even the super security.
Key words: fragile watermarking;chaotic hash function;VQ attack;chaos

摘 要:针对现有分块脆弱水印的缺陷,提出了一种基于混沌 Hash 函数的脆弱水印算法。该方案通过将原始图像分块后做混沌 Hash,生成原始图像的摘要,与由混沌映射生成的混沌二值图像异或后生成二值脆弱水印,置乱加密后嵌入原始图像的 LSB 平面;认证时通过差值图像定位篡改的区域。理论分析和实验仿真表明提出的水印算法不可见性好,能够有效抵抗 VQ 攻击,并能够区分篡改的类型,安全性高。
关键词:脆弱水印;混沌 Hash 函数;VQ 攻击;混沌
DOI:10.3778/j.issn.1002-8331.2008.17.023 **文章编号**:1002-8331(2008)17-0074-04 **文献标识码**:A **中图分类号**:TP391

1 引言

计算机网络技术的迅猛发展和图像编辑软件的日益强大,使得对图像完整性和真实性的验证变得越发重要。近年来,基于脆弱水印的图像认证技术作为一种可以在开放的网络环境下认证来源的完整性技术引起了国内外学者的广泛关注[1-5]。

文献[2]提出了一种新的水印嵌入方式,即图像块的水印嵌入不仅依赖于块本身,还依赖于相邻的块,该方案破坏了水印的分块独立性,可以成功抵抗 VQ 攻击。然而一旦某个图像块被篡改,则与该图像块有关系的所有相邻块均认为被篡改,而且如果篡改的区域较大,检测的结果将是中间的块可以通过认证,而周围相邻的块则认为被篡改;文献[3]提出了一种安全的图像认证算法,该算法基于用户密钥(公钥或私钥)将图像块的高 7 位(7MSBs)、图像索引以及块索引作为 Hash 函数的明文,产生的密文和水印标志异或后作为待嵌入的水印嵌入到原始图像的最低位平面。该算法可以有效抵抗 VQ 攻击,但其主要缺点在于每一个想要认证图像的检测端,图像索引必须作为一个额外的密钥进行传输,而且对于不同的图像,需要不同的索引来表示,从而增加了参数的管理难度,也为发送端和检测端造成了负担;文献[4]提出了一种基于灰度图像的脆弱水印算法,该方案利用灰度图像以及不同的比特移位和分块置乱方法,为不同的图像产生不同的输入密钥,将原始图像和代表该图像的输入密钥经过 Hash 函数作用后,与水印标志异或生成待嵌入的水印,该方法虽能成功抵抗 VQ 攻击,但是通过各种操作生成灰度图像的输入密钥,实现起来较为复杂;文献[5]提出了一种基于矩不变量的脆弱水印方法,该方案将图像矩不变量作为图像的索引,与图像块高 7 位 Hash 值异或生成待嵌入的脆弱水印,该方案同样可以有效抵抗 VQ 攻击,然而图像的矩不变量计算较为复杂,不易于硬件实现,而且传统的 MD5 函数已被攻破,采用该类 Hash 函数实现的脆弱水印算法的安全性也受到了一定质疑。

针对以上种种问题,提出了一种基于混沌 Hash 函数的脆弱水印算法,复合非线性数字滤波器构造的混沌 Hash 函数[6]具有非常高的安全性,而且没有复杂的浮点运算,比现有的 Hash 方法运算速度更快,且更易于扩充和软硬件实现。基于复合非

基金项目:四川省科技公关计划(the Key Technologies R&D Program of Sichuan Province,China under Grant No.05GGG021-026-03)。
作者简介:杨超(1980-),男,硕士,主要研究领域:图像通信;何小海(1964-),男,博士,教授,博士生导师,主要研究领域包括通信与信息处理、图像处理与信息系统,数字通信。
收稿日期:2007-09-12 **修回日期**:2007-12-03

线性数字滤波器构造的混沌 Hash 函数的脆弱数字水印算法,是将原始图像最低位置零后的图像块通过混沌 Hash 函数生成的二值图像与由 Logistic 混沌序列异或生成待嵌入的脆弱水印,置乱加密后嵌入到原始图像的 LSB 位生成含水印图像。检测端通过差值图像来定位篡改。理论分析和实验仿真均表明本文提出的算法不仅可以有效抵抗 VQ 攻击,而且能够区分图像内容篡改或是水印篡改,安全性高,实现简单。

2　混沌Hash 函数

2.1　混沌 Hash 函数的构造

采用文献[6]的算法来产生基于复合非线性数字滤波器(Feedforward-Feedback Nonlinear Filter,FFNF)的混沌 Hash 函数。图 1 和图 2 分别给出了文献[6]中复合非线性数字滤波器和混沌 Hash 函数的生成示意图。具体生成混沌 Hash 函数的过程如下所示:

给定 Hash 值的比特长度 $L\geqslant 128$。首先对原始明文 M 进行填充,使其长度为 L 的多个倍数,然后将 M 分成 L 比特的子块,表示为:$M=(M_1,M_2,\cdots,M_S)$,其中,$M_i=m_i^1 m_i^2 m_i^3\cdots m_i^L$。

步骤 1　输入原始明文序列 M,其中 M 具有 m 个比特,且 $m>0$。

步骤 2　用长度为 n 的比特$(100\ldots0)_2$来对 M 进行填充,使得$(m+n)\bmod L=L/2,1\leqslant n\leqslant L$,其中,$L$ 是偶数。填充之后,M 由 L 比特的子块组成,且每个子块表示为 $M_i,1\leqslant n\leqslant s$。最后一个子块 $M_S=M_S^1\cdots M_S^L$ 是有空余的,更精确地说,第 s 子块的剩余部分 $M_S[M_S^{L/2+1}\cdots\cdots M_S^L]$是空余的。

步骤 3　将第 s 子块的剩余部分扩展,使其长度与原始明文的长度一样,这意味着原始明文的长度小于 $2^{L/2}$。

步骤 4　计算 k 对$\{c_i,s_i\}$系数,使其满足 Kelber 条件,设定初始向量 $H_0=\{0\}_1^L$,密钥 $SK=\{\phi_0,\sigma^{(0)},p_h,p_g\}$,其中,$\phi_0$是初始输入信号,$\sigma^{(0)}$是 FFNF 的初始状态,$p_h$和 p_g分别是 $h(\cdot)$和 $g(\cdot)$的参数。

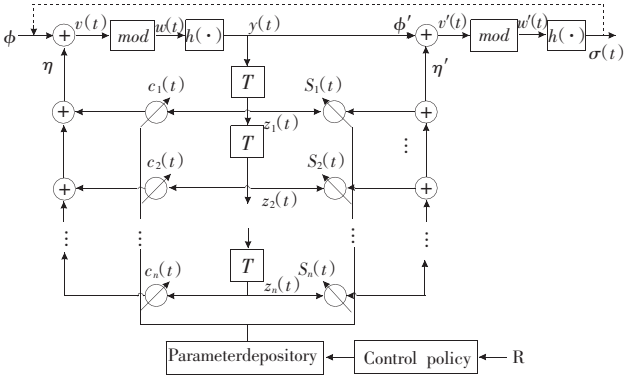

图 1　复合非线性数字滤波器

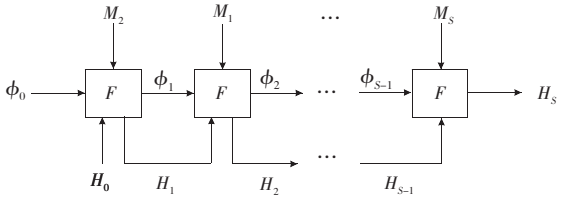

图 2　混沌 Hash 函数的生成

步骤 5　置 $i=1$,对每一个子块执行以下过程:

(1)$\phi=\phi_{i-1}$;

(2)$R=H_{i-1}\oplus M_i=\{r_1,r_2,\cdots,r_L\}$。对于 $j=1:L$,对子块 M_i 进行 CSK 模式调整:

①$q=r_j$ 选择第 q 个子滤波器 $\sigma^{(j)}=\Psi_q(\sigma^{(j-1)},\phi,c_q,s_q)$并进行一步迭代;

②$H_i^j=T_n(\sigma_0^{(j)})$其中 $T_n(\cdot)$是量化函数,表示如式(2):

$$T_n(x)=\begin{cases}1,x\in\bigcup_{d=1}^{2^{n-1}-1}I_{2d}\\0,x\in\bigcup_{d=1}^{2^{n-1}-1}I_{2d+1}\end{cases}\tag{2}$$

其中,$n\in Z^+$,$I_0,I_1,\cdots,I^{2^n-1}$分别表示区间[-1,1]之间 2^n 个相等的部分。

(3)$H_i=H_i^1H_i^2\cdots H_i^L$,$\phi_i=\sigma_0^{(L)}$;

(4)$i=i+1$;执行(1)~(3)直至 $i=s+1$。

步骤 6　输出 Hash 值,$H(M)=H_S=H_S^1H_S^2\cdots H_S^L$。

2.2　混沌 Hash 函数的性能

以 2 维 FFNF 为例,来说明混沌 Hash 函数具有好的散布效果[6]。两对系数和线性映射分为取:

$$\{c_i,s_i\}=\begin{Bmatrix}c_1=[3.57,4],s_1=[-2.30,3]\\c_2=[5.70,7],s_2=[3.70,5]\end{Bmatrix}\tag{3}$$

$$h(w)=g(w)=\begin{cases}\dfrac{w}{p}, & 0\leqslant w<p\\[2mm]\dfrac{w-p}{0.5-p}, & p\leqslant w<0.5\\[2mm]\dfrac{1-w-p}{0.5-p}, & 0.5\leqslant w<1-p\\[2mm]\dfrac{1-w}{p}, & 1-p\leqslant w<1\end{cases}\tag{4}$$

其中,对于 $h(w)$,$p=p_h=0.35$;对于 $g(w)$,$p=p_g=0.25$;密钥 $SK=\{\phi_0,\sigma^{(0)},p_h,p_g\}=\{0.564,0.689,-0.548,0.35,0.25\}$,$L=128$ bit,$H_0=\{0\}_1^{128}$。

为了衡量该混沌 Hash 函数的散列性能,定义了变化比特数。即:对初始明文进行混沌 Hash 变换,然后任意改变初始明文的 1 bit 信息后再进行混沌 Hash,得到另一个混沌 Hash 值,统计两个散列结果相同位置的不同比特个数,称之为变化比特数。理想 Hash 算法的散布效果是初值的细微变化将导致 Hash 结果的每比特都以 50%的概率变化,若散列值长度为 128 bit,则改变明文 1 bit 后的 Hash 结果理想情况下的变化比特数应为 64。图 3 和图 4 统计了该混沌 Hash 函数的变化比特数。

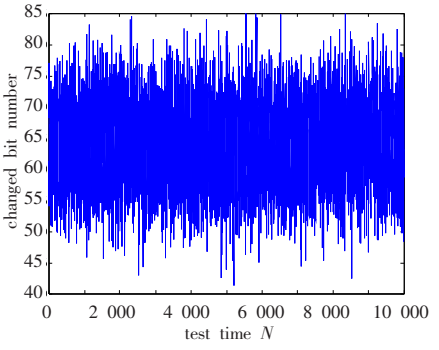

图 3　变化比特数

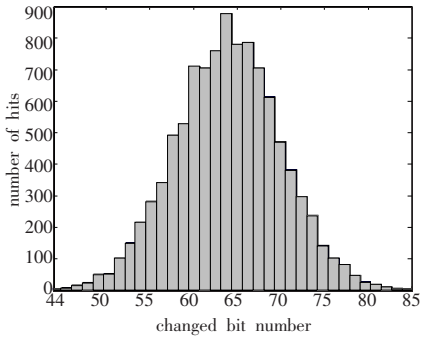

图 4 变化比特数的直方图

由图 3 和 4 可知，在 10 000 次测试下，128 比特散列值的平均比特变化数为 63.89 个，非常接近理想状况下的 64，而且变化比特数都集中在 64 附近，表明该混沌 Hash 函数对明文的散列能力强而且稳定。

3 提出的水印算法

3.1 水印的生成

图 5 给出了水印的生成过程，具体的生成步骤如下：

步骤 1 将原始图像 $I(m\times n)$最低位置零得 I_0，并将图像 I_0 进行 8×8 分块得 B_i，$1\leqslant i\leqslant m\times n/64$。

步骤 2 对每块图像通过混沌 Hash 函数生成摘要，$H_i=chaotic\_Hash(Key_1,B_i)$，其中，$Key_1$表示为混沌 Hash 函数的密钥。

步骤 3 基于密钥 Key_2产生 Logistic 混沌映射 L，并将其分成与混沌 Hash 函数一样长度的段 L_i。

Logistic 混沌映射表示为：

$$L_{q+1}=\alpha L_q(1-L_q) \tag{5}$$

其中，α 表示 Logistic 的混沌系数，密钥 Key_2表示为 Logistic 混沌映射的初值。

步骤 4 对每段混沌 Hash 函数摘要 H_i和每段 Logistic 混沌映射 L_i进行异或：

$$w'=H_i\oplus L_i \tag{6}$$

步骤 5 对各个 w'组合后进行置乱加密，生成待嵌入的脆弱水印信息 w，其中，Key_3是置乱密钥。

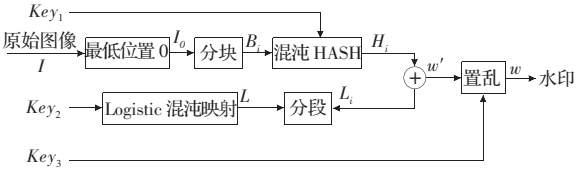

图 5 水印生成示意图

3.2 水印的嵌入

本文采用的水印嵌入算法，是将生成的水印直接嵌入到原始图像的 LSB 位。表示为：

$$I^w=I_0+w \tag{7}$$

其中，I_0表示将原始图像的最低位置零后的图像，I^w表示含水印图像。

3.3 水印的检测

在检测端，采用差值图像进行篡改检测。检测过程示意图如图 6 所示，具体的检测过程如下所示：

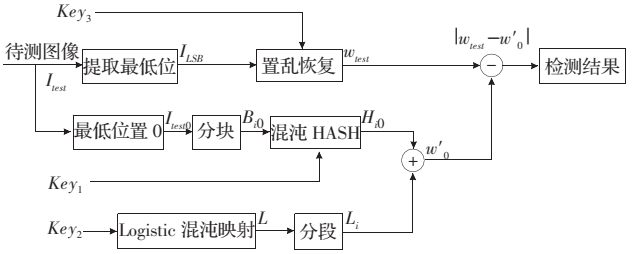

图 6 水印的检测示意图

步骤 1 提取待侧图像的最低位平面，得到 I_{LSB}，并采用与嵌入端相同的密钥 Key_3对其进行置乱恢复，得到恢复后的水印信息 w_{test}；

步骤 2 将待测图像最低位置零，得到 I_{test0}，将 I_{test0}分成 8×8 的块，每块表示为 B_{i0}；进行混沌 Hash 操作：$H_{i0}=chaotic\_Hash(Key_1,B_{i0})$；

步骤 3 基于密钥 Key_2产生 Logistic 混沌映射 L，并将其分成与混沌 Hash 函数一样长度的段 L_i；

步骤 4 对每段 L_i和 H_{i0}进行异或操作，得到：$w'_{0i}=H_{i0}\oplus L_i$，将每部分 w'_{0i}组成 w'_0；

步骤 5 通过检测差值图像$|w_{test}-w'_0|$上的非零点分布判定图像是否被篡改及被篡改的方式和篡改区域。

4 性能分析和实验仿真

4.1 不可见性分析

通过将生成的水印直接嵌入到原始图像的 LSB 平面，可以保持良好的视觉不可感知性。采用峰值信噪比 PSNR（Peak Signal-to-Noise Ratio）来衡量原始图像与含水印图像间的差别。*PSNR* 定义为：

$$PSNR=10\lg\left[\frac{255\times 255}{\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j)-I^w(i,j)]^2}\right] \tag{8}$$

其中，I 代表原始图像，I^w表示含水印图像，m 和 n 分别表示图像的大小。最坏的情况是原始图像和含水印图像的最低位全不相同，此时 *PSNR*=48.130 8 dB，即本水印算法的 *PSNR* 下限是 48.130 8 dB，满足不可见性的要求。

图 7 和图 8 分别给出了原始图像和含水印图像的对比图，从中可以看出含水印图像具有非常好的不可见性。

图 7 原始图像

图 8 含水印图像(PSNR=51.114 1 dB)

4.2 抗 VQ 攻击性能分析

VQ 攻击的前提条件是需要找到：(1)同一图像中的等价类；(2)不同图像间的等价类。提出的水印算法采用了 3 个策略：(1)采用了混沌 Hash 函数，该函数对明文的改变非常敏感，任意 1 bit 的改变，都会使得 Hash 函数的输出有着很大的改变；(2)采用了基于密钥的 Logistic 混沌映射，每块不同的图像对应着不同的 Logistic 混沌序列；(3)采用了置乱加密技术，将混沌 Hash 函数和 Logistic 混沌序列异或后的结果进行置乱加密，破坏了

水印和图像内容间的关系。以上三个策略的应用,有效地破坏了 VQ 攻击的两个前提条件,使得 VQ 攻击不能成功被实施。

4.3　区分篡改性能分析

作为 LSB 位水印，如果不能区分图像内容篡改或是水印篡改,会造成许多误判。比如:在网络中传输一幅含水印图像,攻击者可以以极小的代价来攻击一幅含水印图像的 LSB 位,即攻击水印,使得含水印图像在检测端不能通过检测,只能要求发送端重传;然而,图像内容却并未改变,这就会给发送端和检测端造成许多不必要的重复工作,给网络造成负担。可见,区分图像内容篡改和水印篡改是非常有必要的[7],当含水印图像的内容被篡改时,该算法可以定位出篡改的区域,如果含水印图像中,仅有水印被篡改时,证明图像内容是真实可靠的,不用重传图像,直接利用图像内容即可。下面的一组实验有效说明了该算法区分篡改的性能:

(1)当图像内容不变,只修改嵌入的水印时,检测结果如图 9 和图 10 所示。

图 9　篡改了水印的图像

图 10　差值图像检测结果

从以上实验结果可以看出，当只篡改含水印图像水印时,差值图像中会存在一些类似噪声点,而没有比较集中的点。

(2)当篡改含水印图像内容,保持水印不变时,检测结果如图 11,图 12 所示。

图 11　篡改了内容的水印图像

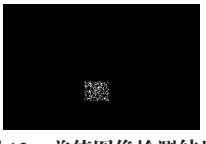

图 12　差值图像检测结果

从图 12 可以看出,当仅篡改含水印图像内容时,差值图像中的非 0 点会集中的显示于被篡改区域,而无随机分布的噪声点。

(3)当同时篡改含水印图像内容和水印时,检测结果如图 13 和图 14 所示。

图 13　篡改含水印图像内容和水印

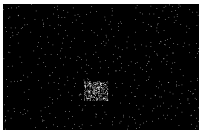

图 14　差值图像检测结果

从以上实验可以看出，同时篡改含水印图像内容和水印时,差值图像中在篡改内容的区域含有集中分布的点,同时也存在着随机分布的噪声点。通过阈值处理法对差值图像进行处理,滤除掉噪声点,只剩下篡改内容的区域,如图 15 所示。

图 15　处理后的检测结果

可见提出的算法不仅可以有效检测出被篡改的区域,而且能够区分图像内容或水印的篡改,增强了水印算法的有效性和实用性。

4.4　水印算法的安全性分析

本文采用了 3 个密钥来保证整个水印算法的安全性。第一个是混沌 Hash 函数的密钥 Key_1，第二个是 Logistic 混沌映射的密钥 Key_2,第三个是置乱加密技术的密钥 Key_3。混沌技术具有众多优点,如:密钥空间大,对初值和系统参数极端敏感,周期长等,仅仅混沌 Hash 函数的密钥就有 $2\times10^{63}\approx2^{210}$[6],再加上另外两个密钥,整个水印算法的安全性将会得到足够的保障。

5　结论

鉴于目前分块脆弱水印存在的问题,提出了一种基于混沌 Hash 函数的脆弱水印算法。该方案充分利用了混沌 Hash 函数良好的散列性能,及混沌映射的随机性好,对初值敏感等的特点。提出的水印算法不可见性好,能够抵抗 VQ 攻击,能区分篡改的类型,安全性高,有一定的实用价值。未来的工作将关注如何提高定位精度和篡改可恢复的问题。

参考文献:

[1] Suthaharan S.Fragile image watermarking using a gradient image for improved localization and security[J].Pattern Recognition Letters,2004(25):1893-1903.

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

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

[4] Suthaharan S.Fragile image watermarking using a gradient image for improved localization and security [J].Pattern Recognition Letters,2004(25):1893-1903.

[5] Liu Fei-long,Wang Yang-sheng.Moment invariants based fragile image watermarking [J].Journal of the Gradute School of the Chinese Academy of Sciences,2004(21):101-107.

[6] Zhang Jia-shu,Wang Xiao-min,Zhang Wen-fang.Chaotic keyed hash function based on feedforward-feedback nonlinear digital filter[J].Physics Letters,2007(362):439-448.

[7] 和红杰,张家树,田蕾.能区分图像或水印篡改的脆弱水印方案[J],电子学报,2005(9):1557-1561.