

文章编号:1001-9081(2012)11-3129-03

doi:10.3724/SP.J.1087.2012.03129

# 基于数独的大容量可篡改定位动态多重水印算法

张 力<sup>1,2\*</sup>,黎洪宋<sup>1</sup>,晏细兰<sup>1</sup>,廉德亮<sup>1</sup>

(1. 深圳大学 信息工程学院, 广东 深圳 518060; 2. 深圳市现代通信与信息处理重点实验室, 广东 深圳 518060)

(\* 通信作者电子邮箱 wzhangli@szu.edu.cn)

**摘要:**提出一种基于数独的大容量动态水印算法,每像素的嵌入容量为2比特。先将原始图像分成大小为 $M \times N$ 像素不重叠区域,把大小为 $2M \times N$ 像素的若干个不同的水印嵌入到每个区域中。若加入水印后图像被篡改,则会造成篡改区域的水印不能正确提取。在任意时刻都可在感兴趣的区域中嵌入任意水印,即嵌入水印的位置和时间是按一定的协议动态确定的。实验结果表明该算法具有较大的水印嵌入容量和较高的篡改定位精确度。

**关键词:**数独;篡改定位;多重水印

**中图分类号:** TP391.41    **文献标志码:**A

## Large-capacity dynamic multiple watermarking with tamperable localization based on Sudoku

ZHANG Li<sup>1,2\*</sup>, LI Hong-song<sup>1</sup>, YAN Xi-lan<sup>1</sup>, LIAN De-liang<sup>1</sup>

(1. Information Engineering College, Shenzhen University, Shenzhen Guangdong 518060, China;

2. Shenzhen Modern Communication and Information Processing Laboratory, Shenzhen Guangdong 518060, China)

**Abstract:** In this paper the large-capacity dynamic multiple watermarking algorithm based on Sudoku was proposed with two bit/pixel embedding capacity. The original image was divided into  $M \times N$  pixels non-overlapping areas. Different watermarks with size of  $2M \times N$  pixels were embedded into the corresponding areas. If the watermarked area was tampered, the watermark in this area could not be extracted correctly. At any time, any watermarks can be embedded into any area as long as the areas are non-overlapping. The experimental results show that the proposed algorithm has a larger watermarking embedding capacity and higher accuracy of tampered localization.

**Key words:** Sudoku; tamper localization; multiple watermarking

## 0 引言

在数字多媒体产品中嵌入水印是一种重要的产权保护手段。水印可以嵌入载体的空间域中,也可以嵌入变换域中,如离散余弦变换(Discrete Cosine Transform, DCT)<sup>[1-3]</sup>、离散小波变换(Discrete Wavelet Transform, DWT)<sup>[4-5]</sup>等。空间域中嵌入水印,可以得到比较大的嵌入容量,而且其嵌入和提取过程比较简单。但是嵌入在空间域中的水印对攻击的鲁棒性相对较差。最低有效位(Least Significant Bits, LSB)算法<sup>[6]</sup>就是通过置换载体中最不重要的位来隐藏水印信息,其原理决定了该方法的嵌入容量很不稳定,对于纹理不够丰富的图像嵌入容量较小。Zhang 等<sup>[7]</sup>在此基础上提出经验模态分解(Empirical Mode Decomposition, EMD)算法,但在嵌入容量上有一定的局限性。Chang 等<sup>[8]</sup>提出基于数独实现的数字水印算法,该算法嵌入容量可达每个像素嵌入 1.5 bit。

本文提出的算法不是采用传统的 $3 \times 3$ 数独矩阵,而是改用 $4 \times 4$ 数独矩阵,且矩阵中的值是从 0 到 15。将水印比特流每 4 个比特分为一组,每组转换成一个十六进制数,得到一个新的十六进制数据流。该十六进制数据流的长度是原来二进制数据流的四分之一。载体中每两个像素可嵌入一个十六进制位,即每个像素的嵌入容量为 2 bit。可在载体的每一个较小的区域中嵌入有意义二值水印图像,若该区域被篡改,则水印将难以正确提取,从而达到了篡改定位的目的。

## 1 提出的算法

本文基于 $16 \times 16$ 的数独矩阵提出了一种大容量动态数

字图像水印算法。水印的嵌入容量可达到 2 比特/像素,水印的检测过程不需要原始图像即可提取嵌入的水印。

### 1.1 数独

数独<sup>[9-11]</sup>的规则是在 $9 \times 9$ 个格子里,每一行与每一列都有 1 到 9 九个数字,每个小九宫格里也有 1 到 9 的九个数字,并且一个数字在每个行列及每个小九宫格里都只能出现一次。

数独最常见的是 $9 \times 9$ 的矩阵,一个 $9 \times 9$ 的数独矩阵有 5472730538<sup>[8]</sup>种不同的组合。若以某个 $9 \times 9$ 数独矩阵为密钥,则其安全性很高,且该密钥所占存储空间很小。本文采用 $16 \times 16$ 的数独矩阵,其组合更为多样,安全性自然更好,而该密钥所占的存储空间只有几百个字节,是可以接受的存储开销。

### 1.2 嵌入过程

本文提出的水印算法实现了多重水印的动态嵌入。水印可以是若干个不同水印,嵌入水印的位置和时间是按一定的协议动态确定的。水印嵌入具体步骤如下:

#### 1) 数独矩阵的产生。

将 $16 \times 16$ 的数独矩阵中的每个数字都减 1,使其数字大小在 0 到 15 的范围内,再利用 Matlab 中的 repmat 函数将 $16 \times 16$ 的数独矩阵复制 $16 \times 16$ 个块,平铺成 $256 \times 256$ 的 M 矩阵,则  $M = \{e_{i,j} \mid 1 \leq i, j \leq 256\}$ , 其中  $e_{i,j} \in \{0, 1, 2, \dots, 8, 9, 10, 11, 12, 13, 14, 15\}$ 。 $M$  矩阵为整个算法的密钥 key1。

#### 2) 水印预处理。

先把二值水印图像转化为含有 1 和 0 的比特流,再将此比特流转化成一个十六进制数据流  $S = \{S_1, S_2, \dots, S_n\}$ ,其中

收稿日期:2012-05-31;修回日期:2012-07-20。基金项目:深圳市互联网产业发展专项资金资助项目(C201005250085A)。

作者简介:张力(1973-),女,山东莱西人,教授,博士,主要研究方向:数字水印、潜信道、信息安全;黎洪宋(1986-),男,湖南常宁人,硕士研究生,主要研究方向:数字水印、信息安全;晏细兰(1985-),女,江西南昌人,硕士研究生,主要研究方向:数字水印、信息安全。

$S_1, S_2, \dots, S_n$  均为从 0 到 15 的数字。

### 3) 原始图像处理。

选取原始图像中大小为  $H \times W$  像素的区域, 将该区域内所有的灰度值都加 1, 然后将其像素矩阵转化成数据流  $I = \{g_1, g_2, \dots, g_{H \times W}\}$ , 其中  $g_1, g_2, \dots, g_{H \times W}$  为图像的各灰度值加 1, 其大小在 1 到 256 之间, 即  $1 \leq g_i \leq 256$ 。灰度图像灰度值大小在 0 到 255 之间, 介于此才将  $M$  矩阵设计成  $256 \times 256$  大小, 并对区域内所有的灰度值都加 1。

### 4) 确定水印嵌入的位置。

从  $I = \{g_1, g_2, \dots, g_{H \times W}\}$  中选取像素对。每个像素对  $(g_i, g_{i+1})$  对应  $M$  矩阵的  $g_i$  行和  $g_{i+1}$  列, 因为  $1 \leq g_i \leq 256$ , 所以  $M(g_i, g_{i+1})$  都能对应到一个十六进制数。到这里不难看出  $M$  矩阵的大小是根据灰度值的大小范围设计出来的, 与图像嵌入区域的大小无关。 $i$  为密钥  $key2$ , 不同的  $i$  值确定了不同的像素对。以  $(g_i, g_{i+1})$  为基点坐标建立  $CE_H, CE_V, CE_B$  三个含 16 个元素的区间, 每个区间的值由 0 ~ 15 组成, 如图 1 所示。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
1								11											
2								12											
3								10											
4								3											
5								7											
6								8											
7								15											
8								5											
9	12	8	1	15	6	7	13	9	2	3	14	5	11	4	0	10	12	8	1
10								9	15	10	7								
11								4	12	6	13								
12								1	6	0	11								
13								14											
14								0											
15								13											
16								6											
17								11											
18								12											
19								10											

图 1  $CE_H, CE_V, CE_B$  区间

### 5) 水印嵌入。

设此时要嵌入  $S$  中的  $S_i$  值, 则在三个区间中必有一个值与之相等, 即:

$$M(x_H, y_H) = M(x_V, y_V) = M(x_B, y_B) = S_i$$

分别计算  $(x_H, y_H)$ 、 $(x_V, y_V)$ 、 $(x_B, y_B)$  与  $(g_i, g_{i+1})$  的城区距离<sup>[9]</sup>:

$$D_H = |g_i - x_H| + |g_{i+1} - y_H|$$

$$D_V = |g_i - x_V| + |g_{i+1} - y_V|$$

$$D_B = |g_i - x_B| + |g_{i+1} - y_B|$$

选取  $D_H, D_V, D_B$  中的最小值  $D_{\min}$ , 其意义为坐标  $(x_{\min}, y_{\min})$  与  $(g_i, g_{i+1})$  的城区距离。令:

$$g_i = x_{\min}$$

$$g_{i+1} = y_{\min}$$

通过修改像素对  $(g_i, g_{i+1})$  的值达到嵌入  $S_i$  的目的。继续嵌入  $S$  中的其他数据, 最终完成整个十六进制数据流的嵌入, 得到嵌入水印的数据流  $Iu$ 。整个嵌入流程如图 2 所示, 水印嵌入算法的流程如图 3 所示。

如果有多个水印需要嵌入, 则选取原始图像中大小为  $H \times W$  像素不重叠区域, 将各水印按以上五个步骤分别嵌入到相应区域中。

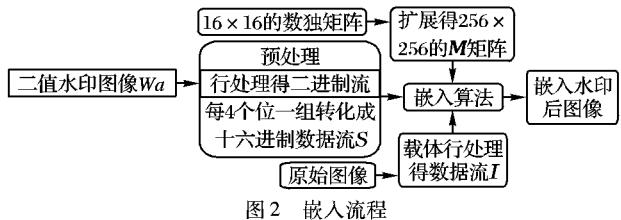


图 2 嵌入流程

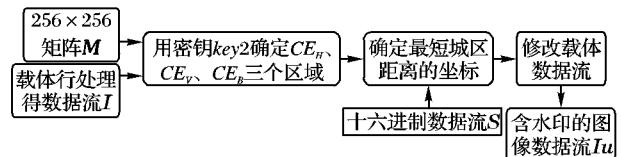


图 3 嵌入算法示意图

### 1.3 提取过程

水印提取过程中不需要用原始图像。提取过程为嵌入过程的逆过程, 即利用密钥  $key1$  和  $key2$  从含水印的载体中提取出水印。水印提取过程流图如图 4 所示, 具体步骤如下:

#### 1) 含水印图像预处理。

选取图像中大小为  $H \times W$  像素嵌入了水印的区域, 将该区域进行行处理得到数据流  $Iw = \{g'_1, g'_2, \dots, g'_{H \times W}\}$ , 其中  $g'_1, g'_2, \dots, g'_{H \times W}$  为含水印图像的灰度值。

#### 2) 获取十六进制数据流。

用密钥  $key2$  获取像素对  $(g'_i, g'_{i+1})$ , 通过获取的像素对可以定位到  $M$  矩阵的相应十六进制数。 $M$  矩阵的信息保存在密钥  $key1$  中, 于是利用密钥  $key1$  得到十六进制数  $S'_i = M(g'_i, g'_{i+1})$  ( $1 \leq i \leq n$ ), 将获取的十六进制数组合成一个十六进制数据流  $S' = \{S'_1, S'_2, \dots, S'_n\}$ 。

#### 3) 获取水印图像。

将  $S'$  转化成二进制数据流  $I'$ , 再将  $I'$  变换成矩阵, 便得到水印图像。



图 4 提取过程的流图

如果有多个嵌入水印需要检测, 则选取图像中嵌入了各个水印的相应区域, 然后按上述三个步骤分别提取嵌入在相应区域中的水印。

## 2 实验结果与分析

实验中采用  $256 \times 256$  像素的原始图像。为体现水印嵌入容量, 采用的水印图像大小是  $512 \times 256$  像素, 载体和水印分别如图 5 和图 6 所示。实验中所采用的数独矩阵如图 7 所示。



图 5 原始图像

把原始图像分成大小为  $8 \times 16$  像素的不重叠的 512 个区域, 用  $L_{mn}$  表示, 说明该区块在整个载体分区的第  $m$  行和第  $n$  列, 其中  $1 \leq m \leq 32, 1 \leq n \leq 16$ 。把原始水印也分成  $16$  像素  $\times 16$  像素不重叠的水印块, 也有 512 个小块, 每个小块分别用  $Z_{mn}$  表示。按本文提出的算法, 将每个水印块分别嵌入到对应下标的载体区域。

中,例如 $Z_{36}$ 嵌入到 $L_{56}$ 中。嵌入水印后的图像如图8所示。嵌入水印后的图8相对原始图像图5的峰值信噪比:PSNR = 46.7640, 可见嵌入的水印对图像质量影响很小。

本文提出了基于数独矩阵的动态多载体水印算法, 利用数独的特性实现超大容量地嵌入密钥, 其嵌入容量可以远大于传统二值水印的容量。那么可嵌入的二值水印图像大小可以是 $16 \times 16$ 像素, 把载体图像大小分成若干个相同或不同水印图像嵌入到每个小块中, 嵌入水印的载体部分区域被修改, 则可以达到篡改定位的目的。从而达到被动防御, 从而达到篡改定位的目的。

图 6 原始水印

1	10	8	4	7	16	12	3	11	2	13	6	9	5	14	0
3	11	14	9	2	5	8	10	12	4	7	0	15	6	13	1
0	13	6	6	14	1	4	11	10	8	9	16	2	12	7	3
5	12	7	2	13	0	9	6	3	5	1	14	8	10	4	11
1	1	15	6	9	13	14	12	7	0	4	2	5	3	10	6
6	5	10	4	11	7	15	8	14	3	1	13	9	2	12	1
12	4	3	12	5	10	0	2	15	11	6	9	7	8	1	14
7	9	14	3	6	1	8	5	13	12	10	0	11	15	4	1
1	8	1	15	6	7	13	9	2	3	14	5	11	14	0	10
4	3	13	5	1	12	11	0	9	15	10	7	14	2	6	8
6	0	10	11	15	3	2	14	4	12	8	13	1	7	9	5
9	14	2	7	8	4	10	5	1	6	0	11	3	13	12	6
10	9	12	13	0	8	6	7	14	1	5	3	4	15	11	2
14	2	6	3	11	9	5	4	0	7	15	12	10	1	8	13
7	5	4	0	12	2	15	1	13	10	11	8	6	14	3	9
8	15	11	1	10	14	3	13	6	9	2	4	12	0	5	7

图 7 采用的数独矩阵



图 8 嵌入水印后图像

对嵌入水印的载体进行如下两种篡改:

- 1) 涂黑部分区域, 如图9所示;
- 2) 嵌入水印的载体图像的左上的书被遮挡, 如图10所示。

图11和图12分别为提取的水印。



图 9 涂黑部分区域



图 10 图片左上被遮挡

图 11 涂黑后提取的水印    图 12 遮挡左上的书提取的水印  
通过实验结果发现, 水印不能正确提取的位置与被篡改

的位置存在很吻合的对应关系, 因而可精确定位到图像被篡改的位置。

从实验结果中可以看出, 本文的算法成功地实现了多重水印大容量的动态嵌入与提取, 同时也实现了较为精确的篡改定位。如表1所示, 提出的算法相对LSB以及EMD算法有一定的优势。所以本文的算法充分体现了数独在水印嵌入算法中的优越性, 较Chang对数独的应用有进一步的拓展。

表 1 嵌入算法定性比较

水印嵌入算法	水印嵌入容量	水印的安全性	嵌入水印后载体图像的质量
本文算法	2 bit/pixel	较高	较好(PSNR > 35)
LSB 嵌入算法	$\leq 2$ bit/pixel	一般	不稳定
EMD 嵌入算法	$\leq 2$ bit/pixel	一般	不稳定

LSB和EMD的嵌入容量与原始载体纹理相关, 纹理丰富的嵌入容量较高, 但一般取灰度值低两位嵌入, 在保证原始载体质量的情况下其嵌入容量一般小于2bit/pixel; 本文提出的算法以数独矩阵为密钥, 因为数独有很多组合形式, 从而使密钥库丰富, 提高了嵌入水印的安全性。其他两种算法嵌入位置容易被穷举方式破解, 安全性较一般; 对于纹理较丰富的原始载体, 三种算法都较好。但是对于纹理比较平滑的载体图像, 若LSB和EMD嵌入算法的嵌入容量为2 bit/pixel, 则图像质量较差。

### 3 水印算法性能分析

本文所提出的水印算法具有大容量和篡改定位的特点, 而且水印的嵌入时间和嵌入内容是根据作者需求随时嵌入进行的。

#### 3.1 嵌入容量

本文所提出的算法具有嵌入容量大的优点。假设要嵌入的二值水印大小是 $M \times N$ 像素, 相应二进制比特流长度是 $M \times N$ 。转化成十六进制数据流后其长度只有原来的四分之一。每嵌入一个十六进制数需要两个像素, 用这两个像素的灰度值来确定映射到数独矩阵的坐标, 也就是说每2个像素能嵌入4个比特, 即每像素能嵌入2比特。

#### 3.2 篡改定位

因为该算法的嵌入容量大, 因此能在在一个很小的区域中嵌入有意义的二值图像。例如可以在 $8 \times 16$ 像素的小区块里嵌入一个 $16 \times 16$ 像素的二值水印图像。一幅图像作品可以选取很多个小区域来嵌入水印。在嵌入区域中提取出的水印是混乱的没意义的二值图像时, 可以判断该区域被篡改过, 例如在第2章中部部分区域提取出的水印是没意义的图像, 就可以认定该区域被篡改了。综合实验结果与分析, 可以得出该算法能精确到较小区块的篡改定位<sup>[12]</sup>。

#### 3.3 水印的动态嵌入

一幅图像作品的多个作者在任意时刻可以根据各自的兴趣在不同区域中嵌入水印信息, 即嵌入时刻是任意的, 这是一个动态嵌入的过程。当然, 为防止区域选择的重叠, 必须按一个已定的嵌入协议进行水印动态嵌入。所以该算法充分利用大容量的优点很好地实现了动态多重水印的嵌入。

### 4 结语

本文提出的基于 $16 \times 16$ 数独矩阵的算法, 不仅在嵌入容量上有大的突破, 而且实现了动态多重水印的嵌入与较为精确的篡改定位。实验结果表明, 该算法效果显著、可行性好。但本文提出的算法尚属于脆弱水印, 下一步工作主要是提高该算法的鲁棒性。

(下转第 3146 页)

有效的。但是,当前在本文协议中没有考虑恶意的参与者,恶意参与者的最大利益不是获得秘密,而是阻止他人获得秘密,今后,将进一步研究如何防止恶意参与者的解决方案。

#### 参考文献:

- [1] SHAMIR A. How to share a secret [J]. Communications of the ACM, 1979, 22(11): 612–613.
- [2] BLAKELEY G R. Safeguarding cryptographic keys [C]// Proceedings of the National Computer Conference. New York: AFIPS Press, 1979: 313–317.
- [3] CHOR B, GOLDWASSER S, MICALI S. Verifiable secret sharing and achieving simultaneity in the presence of faults [C]// 26th Annual Symposium on Foundations of Computer Science. Washington, DC: IEEE Computer Society, 1985: 383–395.
- [4] FELDMAN P. A practical scheme for non-interactive verifiable secret sharing [C]// FOCS' 87: 28th IEEE Symposium on Foundations of Comp. Science. Washington, DC: IEEE Computer Society, 1987: 427–437.
- [5] PEDERSEN T P. Distributed provers with applications to undeniable signatures [C]// Proceedings of Eurocrypt' 91, LNCS 547. Berlin: Springer, 1991: 221–238.
- [6] LIN H Y, HARN L. Fair reconstruction of a secret [J]. Information Processing Letters, 1995, 55(1): 45–47.
- [7] HALPERN J, TEAGUE V. Rational secret sharing and multiparty computation [C]// Proceedings of the 36th Annual ACM Symposium on Theory of Computing. New York: ACM, 2004: 623–632.
- [8] KOL G, NAOR M. Cryptography and game theory: Designing protocols for exchanging information [C]// Proceedings of the 5th Theory of Cryptography Conference. Berlin: Springer, 2008: 317–336.
- [9] MALEKA S, AMJED S, RANGAN C P. Rational secret sharing with repeated games [C]// 4th Information Security Practice and Experience Conference, LNCS 4991. Berlin: Springer, 2008: 334–346.
- [10] MALEKA S, AMJED S, RANGAN C P. The deterministic protocol for rational secret sharing [C]// the 22 th IEEE International Parallel and Distributed Processing Symposium. Washington, DC: IEEE Computer Society, 2008: 3651–3657.
- [11] IZMALKOV S, LEPINSKI M, MICALI S. Verifiably secure devices [C]// TCC 2008: 5th Theory of Cryptography Conference, LNCS 4948. Berlin: Springer, 2008: 273–301.
- [12] MICALI S, SHELAT A. Purely rational secret sharing [C]// TCC 2009: 6th Theory of Cryptography Conference, LNCS 5444. Berlin: Springer, 2009: 54–71.
- [13] TOSHIYHKO I, KOICHIRO W, KEISUKE T. A rational secret - sharing scheme based on RSA-OAEP [J]. IEICE Transactions on Fundamentals of Electronics Communications and Computer Science, 2010, E93-A(1): 42–49.
- [14] KATZ J. Bridging game theory and cryptography: Recent results and future directions [C]// TCC 2008: 5th Theory of Cryptography Conference, LNCS 4984. Berlin: Springer, 2008: 251–272.
- [15] ABRAHAM I, DOLEV D, GONEN R, et al. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation [C]// 25th ACM Symposium Annual on Principles of Distributed Computing. New York: ACM, 2006: 53–62.
- [16] MICALI S, RABIN M, VADHAN S. Verifiable random functions [C]// Proceedings of the 40th IEEE Symposium on Foundations of Computer Science. Washington, DC: IEEE Computer Society, 1999: 120–130.
- [17] DODIS Y, YAMPOLSKIY A. A verifiable random function with short proof and keys [C]// 8th International Workshop on Theory and Practice in Public Key Cryptography, LNCS 3386. Berlin: Springer, 2005: 416–431.

(上接第 3131 页)

#### 参考文献:

- [1] XIA JUNJUN, PANETTA K, AGAIAN S. Color image enhancement algorithms based on the DCT domain [C]// SMC: 2011 IEEE International Conference on Systems, Man, and Cybernetics. Washington, DC: IEEE Computer Society, 2011: 1496–1501.
- [2] AGARWAL C, MISHRA A, SHARMA A. Digital image watermarking in DCT domain using fuzzy inference system [C]// CCECE: 24th Canadian Conference on Electrical and Computer Engineering. Washington, DC: IEEE Computer Society, 2011: 000822 – 000825.
- [3] DUBOLIA R, SINGH R, BHADORIA S S, et al. Digital image watermarking by using discrete wavelet transform and discrete cosine transform and comparison based on PSNR [C]// 2011 International Conference on Communication Systems and Network Technologies. Washington, DC: IEEE Computer Society, 2011: 593–596.
- [4] GHORBANI M, FIROUZMAND M, FARAAHI A. DWT-DCT (QCD) based copy-move image forgery detection [C]// 18th International Conference on Systems, Signals and Image Processing. Washington, DC: IEEE Computer Society, 2011: 1–4.
- [5] LIN QIWEI, TANG JISHENG, WU XUFENG. A new DWT & multi-strategy watermark embedding algorithm [C]// 2011 IEEE International Conference on Anti-Counterfeiting, Security and Identification. Washington, DC: IEEE Computer Society, 2011: 57–60.
- [6] ZHANG JUN, COX I J, DOERR G. Steganalysis for LSB matching in images with high-frequency noise [C]// 9th IEEE Workshop on Multimedia Signal Processing. Washington, DC: IEEE Computer Society, 2011: 1–4.
- [7] ZHANG X, WANG S. Efficient steganographic embedding by exploiting modification direction [J]. IEEE Communications Letters, 2006, 10(11): 1–3.
- [8] CHANG C-C, CHOU Y-C. An information hiding scheme using Sudoku [C]// 3rd International Conference on Innovative Computing Information and Control. Washington, DC: IEEE Computer Society, 2008: 17.
- [9] WU YUE, NOONAN J P, AGAIAN S. Image encryption using the rectangular Sudoku cipher [C]// ICSSE: International Conference on System Science and Engineering. Washington, DC: IEEE Computer Society, 2011: 704–709.
- [10] ZOU YANG, TIAN XIAOLIN, XIA SHAOWEI, et al. A novel image scrambling algorithm based on Sudoku puzzle [C]// CISP: 4th International Congress on Image and Signal Processing. Washington, DC: IEEE Computer Society, 2011: 737–740.
- [11] NAINI P M, FAKHRAIE S M, AVANAKI A N. Sudoku bit arrangement for combined demosaicing and watermarking in digital camera [C]// DBKDA: Second International Conference on Advances in Databases Knowledge and Data Applications. Washington, DC: IEEE Computer Society, 2010: 41–44.
- [12] LIU ZHAOQING, LI QIONG, ZHANG HUI, et al. An image structure information based robust hash for tamper detection and localization [C]// IIH-MSP: 2010 Sixth International Conference on Digital Object Identifier Intelligent Information Hiding and Multimedia Signal Processing. Washington, DC: IEEE Computer Society, 2010: 430–433.