

基于动态差分扩展的强鲁棒数据库水印算法研究

汪天琦^{1,2,3}, 张迎周^{1,2,3}, 邱云龙^{1,2,3}, 李鼎文^{1,2,3}, 朱林林^{1,2,3}

(1. 南京邮电大学计算机学院, 江苏 南京 210023; 2. 南京邮电大学软件学院, 江苏 南京 210023;

3. 南京邮电大学网络空间安全学院, 江苏 南京 210023)

摘要: 科技行业的快速发展带来信息量的暴增, 各行各业都需要收集和应用大量的数据, 海量数据在发挥价值的同时, 给数据安全领域带来了史无前例的挑战。关系型数据库作为数据的底层存储载体之一, 其存储的数据规模大、数据内容丰富、数据隐私度高。数据库的数据一旦泄露将会造成巨大的损失, 保护数据库的所有权, 确认数据的归属刻不容缓。对于现有的数据库水印技术来说, 提高水印嵌入容量和减小数据失真之间存在固有矛盾问题, 为了缓解此问题且进一步提高水印的鲁棒性, 提出了一种基于动态差分扩展的强鲁棒数据库水印算法。该算法选取QR码作为水印, 利用经过Haar小波变换的图像低频部分进行奇异值分解(SVD, singular value decomposition), 提取部分特征值, 用取余后的特征值作为待嵌入的水印序列, 使得相同长度的水印序列包含更多信息, 缩短了嵌入水印的长度。该算法结合自适应差分进化算法和最小差值算法选择最佳嵌入属性位, 以缓解传统差分扩展技术在嵌入水印时计算效率低、数据失真大、鲁棒性差的问题, 提高水印嵌入容量的同时减少了数据的失真。实验结果表明, 该算法保证高水印嵌入率的同时数据失真较低, 能够抵御多种攻击, 具有良好的鲁棒性, 追踪溯源的能力强, 且与现有的算法对比优势明显, 在数据安全领域具有广阔的应用前景。

关键词: 数据库水印; 差分进化; 差分扩展; SVD; Haar小波变换; QR码

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2023065

Research on strong robustness watermarking algorithm based on dynamic difference expansion

WANG Tianqi^{1,2,3}, ZHANG Yingzhou^{1,2,3}, DI Yunlong^{1,2,3}, LI Dingwen^{1,2,3}, ZHU Linlin^{1,2,3}

1. Department of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

2. Department of Software, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

3. Department of Cyberspace Security, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract: A surge in the amount of information comes with the rapid development of the technology industry. Across all industries, there is a need to collect and utilize vast amounts of data. While this big data holds immense value, it also poses unprecedented challenges to the field of data security. As relational databases serve as a

收稿日期: 2022-04-02; 修回日期: 2022-11-21

通信作者: 张迎周, zhangyz@njupt.edu.cn

引用格式: 汪天琦, 张迎周, 邱云龙, 等. 基于动态差分扩展的强鲁棒数据库水印算法研究[J]. 网络与信息安全学报, 2023, 9(5): 150-165.

Citation Format: WANG T Q, ZHANG Y Z, DI Y L, et al. Research on strong robustness watermarking algorithm based on dynamic difference expansion [J]. Chinese Journal of Network and Information Security, 2023, 9(5): 150-165.

fundamental storage medium for data, they often contain large-scale data rich in content and privacy. In the event of a data leak, significant losses may occur, highlighting the pressing need to safeguard database ownership and verify data ownership. However, existing database watermarking technologies face an inherent tradeoff between improving watermark embedding capacity and reducing data distortion. To address this issue and enhance watermark robustness, a novel robust database watermarking algorithm based on dynamic difference expansion was introduced. The QR code was employed as the watermark, the SVD decomposition of the low frequency part of the image was utilized after Haar wavelet transform. By extracting specific feature values and using residual feature values as the watermark sequence, it was ensured that the same-length watermark sequence contains more information and the embedded watermark length can be reduced. Furthermore, by combining the adaptive differential evolution algorithm and the minimum difference algorithm, the optimal embedding attribute bits were selected to alleviate the problems of low computational efficiency, high data distortion and poor robustness of traditional difference expansion techniques in embedding watermarks, and to improve the embedding capacity of watermarks while reducing the distortion of data. Experimental results demonstrate that the proposed algorithm achieves a high watermark embedding rate with low data distortion. It is resilient against multiple attacks, exhibiting excellent robustness and strong traceability. Compared to existing algorithms, it offers distinct advantages and holds great potential for broad application in the field of data security.

Keywords: database watermarking, differential evolution, difference expansion, singular value decomposition, Haar wavelet transform, quick response code

0 引言

科技行业的快速发展带来的是信息量的暴增，各行各业都需要收集和应用大量的数据，累积的数据规模正以惊人的速度增长。海量数据蕴含的商业价值巨大，但其发挥价值的同时，也为数据安全领域带来了史无前例的挑战。数据库作为最常用的底层存储工具之一，其信息安全问题不容小觑。国内外学者大多是通过数据库水印技术来保护数据库的数据安全，解决信息泄露、源头难以追踪的困扰。在数据库背后利益的驱使下，攻击者在数据库所有者不知情的情况下对其加以复制、分发和篡改。数据库的数据一旦泄露将会造成巨大损失，保护数据库的所有权，确认数据的归属刻不容缓。

数字水印技术^[1]可以有效解决很多信息安全问题，作为一种安全有效的保护数字版权的技术手段，数字水印技术广泛应用于多媒体领域^[2-3]。但是，针对数据库的数字水印研究相对较少。目前，数字水印技术主要关注其不可见性、鲁棒性、安全性和嵌入容量等特性^[4]。

可逆水印技术是数字水印技术的一个重要分支，Zhang 等^[5]提出了经典的可逆水印算法，该算法利用直方图平移拓展出的冗余空间按照指定

的运算公式来嵌入水印，经过逆运算即可恢复出原始数据和水印信息，但是该算法的鲁棒性较差，在面对不法分子的恶意攻击时提取出的水印信息不完整。针对 Zhang 等提出算法的不足，Hu 等^[6]引入了遗传算法训练出最佳的水印嵌入位置，结合分组密钥和平移直方图设计了一种基于遗传算法的直方图平移可逆水印技术。

对于数据库水印来说，水印隐藏在关系数据库中，数据值变化较大，严重破坏原始数据的质量，水印的鲁棒性低，攻击者很容易破坏水印信息。Gupta 等^[7]吸取了数字水印技术的经验，提出了基于差分扩展的可逆数据库盲水印，选取元组中的任意两个属性值，通过差分扩展计算将水印嵌入数值中，对两个属性值进行计算嵌入一位水印，导致数值改动大于水印的嵌入，影响了数据库的可用性和鲁棒性。差分扩展技术中增加了水印嵌入容量，但不可避免地增大了数据的失真，于是研究者想到利用优化算法改进差分扩展数据库水印方法。Jawad 等^[8]利用遗传算法改进了基于差分扩展的可逆数据库盲水印，提出基于遗传算法的差分扩展可逆数据库水印，提高数据库水印算法的鲁棒性同时减少数据失真。但是，经过遗传算法优化的水印嵌入率并不尽如人意。邱升红^[9]

设计了一种基于差分扩展和人工蜂群算法的数据库可逆水印方法,使用人工蜂群算法代替遗传算法进行优化,增大了水印的嵌入容量并减少了数据的失真,进一步提高了水印算法的鲁棒性。宋岩等^[10]利用改进的布谷鸟算法启发式地搜索水印嵌入属性位,通过差分扩展嵌入水印,对于大规模的数据库算法的水印嵌入率较高,对抗攻击的能力较强。孔嘉琪等^[11]基于模拟退火改进的粒子群算法寻找更好的水印嵌入位置,提出了基于属性重要度的带权损失函数,解决了现有方案在抗属性维度攻击时鲁棒性较差的问题。由于数据库存储的是纯数据,其数值的改变对原载体的影响远大于多媒体水印,所以数据库水印技术很难在水印容量和数据失真方面保持平衡。Shi 等^[12]试图通过恢复原始数据和嵌入式水印信息来克服数据失真的问题。可逆水印技术能够在提取水印的同时恢复数据,然而目前的可逆数据库水印技术在嵌入水印后,数据质量和水印鲁棒性方面存在缺陷,基于此,Ge 等^[13]对水印容量和数据失真做了研究和改进,在一定的数据失真范围内提高了水印嵌入容量。现有的技术主要通过增加水印长度或提取水印特征压缩数据两种方法来达到提高水印嵌入容量的目的。陈青等^[14]提出了一种新的基于旋转稳定区域和两级奇异值分解(SVD, singular value decomposition)的水印算法,保证水印的嵌入容量,且具有良好的不可见性和较高的鲁棒性。关虎等^[15]将大容量、高容错的二维码理论应用于变换域图像水印算法,在保证水印不可见性和算法安全性的基础上,显著提高了水印容量和鲁棒性。在学者不断研究改进的过程中,数据库水印的鲁棒性和嵌入容量逐渐提高,数据的失真能够控制在合理的范围内。

1 本文算法

1.1 算法框架

大多数数据库水印技术是单纯地嵌入一串数字序列或一副图像,包含的信息量较少且容易被攻击者破坏。当数据库发生泄露被恶意攻击时常常会导致水印被破坏,无法根据提取的信息锁定攻击者,因此水印的鲁棒性和溯源度较差。为了解决这一问题,需要增加水印的嵌入容量,但是嵌

入容量过多会造成数据严重失真,影响数据库正常使用。针对数据库冗余性小、水印嵌入容量和数据失真之间存在固有矛盾等问题,本文提出了一种基于自适应差分进化(ADE, adaptive differential evolution)和动态差分扩展(DDE, dynamic difference expansion)的强鲁棒数据库水印算法,以下简称 ADE-DDEW。所提算法首先对经过 Haar 离散小波变换的图像低频部分进行 SVD,提取非 0 特征值,然后对特征值进行取余,对取余后的特征值进行动态压缩作为待嵌入的水印序列。结合自适应差分进化算法和最小差值算法选择最佳的水印嵌入属性位,最后通过动态地选取传统的或改进的差分扩展技术进行水印的嵌入。算法的总体框架如图 1 所示,主要包括水印预处理、水印嵌入和水印提取 3 个部分。

1.2 数据库水印预处理

为了解决嵌入容量和数据失真之间的矛盾,尽可能达到嵌入更多身份信息的同时压缩待嵌入水印长度的目的,将水印嵌入数据库之前,需要对水印图像进行预处理。为了包含更多的信息,图像水印是最佳的选择。图像作为水印嵌入数据库往往采用分割的方法,未经过数据压缩直接嵌入导致有效嵌入率很低。QR 码容量密度大且容错性极强,因此使用 QR 码作为原始的水印载体。假设关系型数据库 $DB(PK, Arr_1, \dots, Arr_i, \dots, Arr_N)$, 其中 PK 为数据库每个元组的主键, Arr_i ($1 \leq i \leq N$) 为 DB 的第 i 个属性列, DB 中包含 M 个元组,记为 $\{t_1, \dots, t_j, \dots, t_M\}$, 每个元组 t 都有唯一的主键 $t.PK$ 和 N 个属性列,第 j 个元组的第 i 列属性值 Arr_i 值表示为 $t_j.Arr_i$, 数据库 DB 的第 i 列属性值表示为 $\{t_1.Arr_i, \dots, t_j.Arr_i, \dots, t_M.Arr_i\}$ 。待嵌入的 QR 码分别为嵌入数值信息的 QR 码和嵌入文本信息的 QR 码。QR 码的预处理过程主要分为频域分解、SVD 和动态压缩 3 个阶段,具体操作如下。

基于 Haar 小波变换的图像压缩算法的核心思想是通过计算平均值和差值得到细节系数(分为低频和高频),将图像分解成若干个高频图像和一个低频图像,可以在不影响主要信息的情况下初步对原始图像进行压缩,达到减小图像信息量的目的。Haar 能够完全不失真地还原出原始图像,进而提高水印的可溯源性。本文采用 Haar 小波变

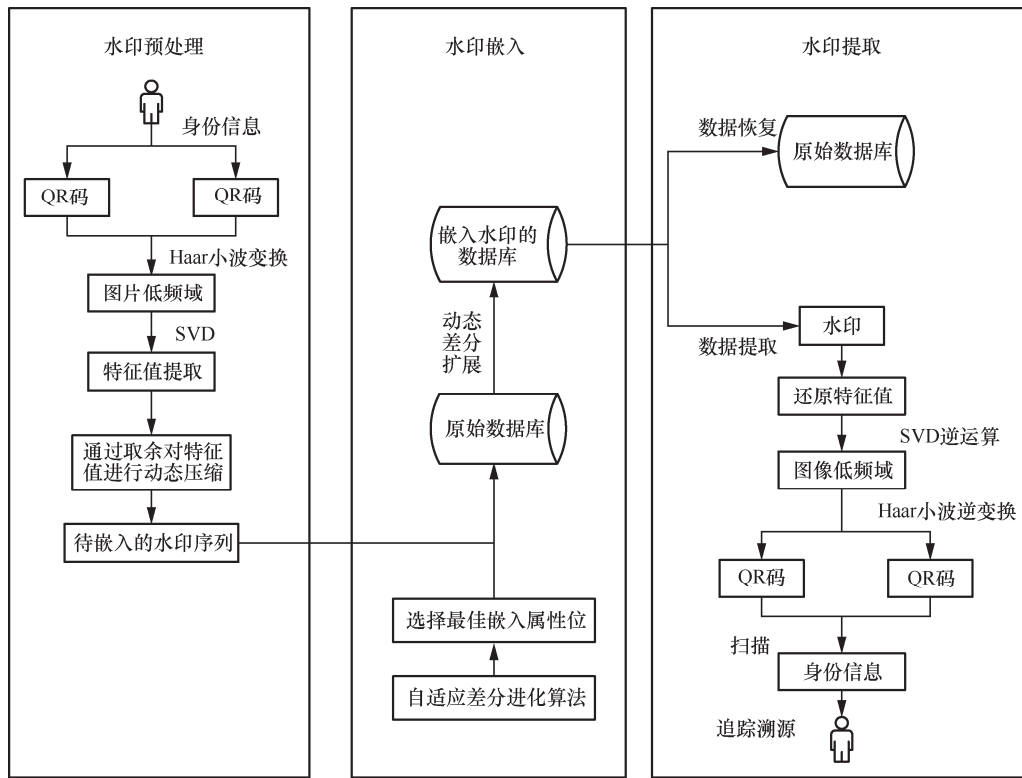


图 1 算法的总体框架

Figure 1 General framework of the algorithm

换对 QR 码图像进行频域分解，由于低频部分包含图像大多数的有用信息，并且低频信号对于图像压缩、高斯噪声等多种图像处理操作有很强的抵御能力，所以本文提取分解后的低频图像作为处理完成的水印图像。假设水印图像的 4 个相邻像素值是 $[a, b, c, d]$ ，基于 Haar 小波变换的 QR 码频域分解的具体步骤如下。

步骤 1 将 4 个像素值两两分组，得到 $[a, b]$ 和 $[c, d]$ 。

步骤 2 根据式(1)分别计算这两组像素值的均值和差值，很容易得到两组均值和差值分别是 $[I_{1,0}, II_{1,0}]$ 和 $[I_{1,1}, II_{1,1}]$ ，其中 I 表示均值， II 表示差值。将这两组值重新组合成为 $[I_{1,0}, I_{1,1}, II_{1,0}, II_{1,1}]$ ，这两个均值被称为一级均值。

步骤 3 分别计算两个一级均值（即一级均值 $[I_{1,0}, I_{1,1}]$ ）的差值和均值，根据式(2)可以进行二级小波系数的计算，其均值和差值分别是 $I_{0,0}$ 和 $II_{0,0}$ 。用 $I_{0,0}$ 和 $II_{0,0}$ 替换掉步骤 2 的两个一级均值，就可以得到 4 个相邻像素值 $[a, b, c, d]$ 最终的二级 Haar 小波变换系数 $[I_{0,0}, II_{0,0}, II_{1,0}, II_{1,1}]$ 。

$$\begin{cases} I_{1,0} = \frac{a+b}{2} \\ I_{1,1} = \frac{c+d}{2} \\ II_{1,0} = \frac{a-b}{2} \\ II_{1,1} = \frac{c-d}{2} \end{cases} \quad (1)$$

$$\begin{cases} I_{0,0} = \frac{I_{1,0} + I_{1,1}}{2} \\ II_{0,0} = \frac{I_{1,0} - I_{1,1}}{2} \end{cases} \quad (2)$$

步骤 4 根据实际的需求重复步骤 2、步骤 3，最后提取由低频系数 $I_{0,0}$ 组成的水印低频图像。

SVD^[16]主要应用在数据降维和数据压缩领域，是很多机器学习算法的基石，其核心思想是将图像矩阵分解出特征值和特征向量。矩阵的运算可以描述为线性空间中的变换，当一个矩阵进行运算时，本质上就是在矩阵空间下进行的一次线性变换（拉伸或者旋转），线性变化无法直观地通过图像来展示，但是经过奇异值分解得到的部分特

征向量代表了该矩阵的主要变化方向。在这些方向上进行变换,就可以模拟矩阵的运算。基于上述性质,SVD可以做到无失真的压缩,不对图像造成损坏的同时通过逆运算还原出原始图像。

目前,图像作为水印嵌入数据库的场景基本是需要分割图像,然后将分割后的子图像分别嵌入分组后的子数据表。这种传统的方法不仅对图像的尺寸有一定的要求,而且由于数据库的分组不是无限的,所以嵌入的图像也有一定的限制。为了将二维的图片水印压缩成一维的水印序列,使用 SVD 技术对频域分解后提取出的低频图像进一步压缩,提取部分特征值生成水印序列。SVD 不要求图像的尺寸并且有很强的数据降维能力,提高了水印信息的压缩率,使得相同长度的水印序列中包含更多的数据信息。

假设 QR 码图像是一个 $M \times N$ 的矩阵 A , 经过 SVD 可以变成如式(3)的形式。

$$A = USV^T \quad (3)$$

其中, $U \in \mathbf{R}^{M \times M}$ 是左奇异值矩阵, $V \in \mathbf{R}^{N \times N}$ 是右奇异值矩阵, $S \in \mathbf{R}^{M \times N}$ 是奇异值矩阵, 仅在对角线上有值, 其他元素均为 0, 其一般形式如式(4)所示。

$$S = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \sigma_N & 0 \end{bmatrix}_{M \times N} \quad (4)$$

式中 $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_N)$ 是一个对角矩阵, $\{\sigma_1, \sigma_2, \dots, \sigma_N\}$ 表示图像的特征值, 大小满足 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k = \dots = 0$, 呈速度下降趋势。第一个特征值 σ_1 远大于其他的特征值, 包含最多的数据信息, 以此类推, 越末尾的特征值越小, 含信息量越低。基于这一特性, 提取出的水印序列即使丢失部分特征值也不影响水印图像的还原, 鲁棒性较高。

水印 SVD 流程如图 2 所示。为了增加嵌入的水印信息, 使其可以更好地抵御攻击, 分别选取嵌入数值信息和文本信息的 QR 码低频图像进行 SVD, 首先对两幅低频图像 P_1 和 P_2 进行 SVD, 分别得到特征值矩阵 S_1 和 S_2 , 然后从 S_1 和 S_2 中提取非 0 特征值得到一维水印序列 $W(\sigma_1, \sigma_2, \dots,$

$\sigma_k), 1 \leq k \leq (M + N)$ 。

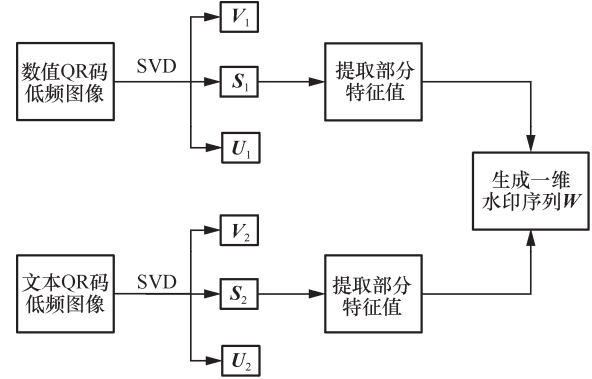


图2 水印 SVD 流程

Figure 2 Watermark singular value decomposition flow

1.3 算法设计

(1) 数据库分组

大多水印算法的分组采用传统的哈希分组方法, 利用主键和密钥作为参数进行分组。考虑到差分扩展的计算特性, 为了尽可能地减小数据的失真, 本文采用聚类的分组方法, 将相似的数据分到一组, 聚类的方法可以避免从遭受置乱攻击的数据库中提取出来的水印乱序的问题。根据 K-means 算法^[17]对数据库 DB (PK, Arr₁, ..., Arr_i, ..., Arr_N) 进行分组, 将属性数值较为相似的元组分为一组, 划分为 K 个大小不同的子组 {G₁, G₂, ..., G_K}。为了提高水印的鲁棒性, 采用冗余嵌入水印的方法, 即对于每个子组来说, 其待嵌入水印的元组都嵌入相同的水印比特位, 因此要求分组数量 K=L, L 为要嵌入的水印序列的长度。

定义 1 属性失真: 定义数据库嵌入水印后大于原本属性列最大值或者小于最小值的属性值为失真的属性值。即对于有 N 个属性列的数据库 DB 来说, 每个特征属性列数值的非失真范围为 [Arr_i^{min}, Arr_i^{max}](i=1, 2, ..., N)。

定义 2 水印嵌入容量 (WEC, watermark embedding capacity): 把嵌入水印的元组数和总元组数的比值定义为水印嵌入容量。

由于水印嵌入在数据库的属性值中, 因此水印的嵌入容量与嵌入水印位置的个数有关, 即与数据库中嵌入水印的元组数相关。水印嵌入容量的计算方法如式(5)所示。其中, Tuple 表示数据

库分组后子组中的元组总数, Tuple_{ew} 表示嵌入水印后属性值未超过失真范围的元组行数。

$$\text{WEC} = \frac{\text{Tuple}_{ew}}{\text{Tuple}} \quad (5)$$

定义3 数据失真量 (AoDD, amount of data distortion): 定义嵌入水印前后属性值的改变率为数据失真量。即对于任意属性列来说, 是嵌入水印造成的整体数据差值和该属性列极差之间的比值。

数据的变化量受原始数值的影响较大。具体来说, 一位数的数据变化量只能在区间[0,9], 而两位数则可以达到[0,99], 以此类推, 可以得出原始数值越高, 可能导致数据变化量越大, 如果只是单纯地累加数值变化量, 对于有失真统计来说有失公平。本文数据失真量的计算方法如式(6)所示。

$$\text{AoDD} = \frac{1}{\text{Col}} \sum_{i,j=1}^{\text{Col}} \frac{1}{E_{S_i}} \cdot \sum_{t_r \in S_i, S_j} \frac{|t_r \cdot \text{Arr}'_i - t_r \cdot \text{Arr}_i|}{\text{Arr}_i^{\max} - \text{Arr}_i^{\min}} + \frac{|t_r \cdot \text{Arr}'_j - t_r \cdot \text{Arr}_j|}{\text{Arr}_j^{\max} - \text{Arr}_j^{\min}} \quad (6)$$

其中, Col 表示数据库要嵌入水印的属性列个数, S_i 表示第 i 列属性中嵌入水印后的属性值集合, E_{S_i} 表示 S_i 中有多少个元素, $\text{Arr}_i^{\max}$ 表示第 i 列属性中的最大值, $\text{Arr}_i^{\min}$ 表示第 i 列属性中的最小值, $t_r \cdot \text{Arr}_i$ 表示第 r 行第 i 列的属性值, $t_r \cdot \text{Arr}'_i$ 表示第 r 行第 i 列嵌入水印后的属性值。由于差分扩展技术使每个元组改变两个属性数值, 因此要分别对其改变量进行计算。

将上述 WEC 和 AoDD 两个指标作为目标函数的主体得到最终的目标函数, 如式(7)所示。

$$\text{Func}_{\text{obj}} = w_c \cdot \text{WEC} + w_d \cdot \text{AoDD} \quad (7)$$

其中, w_c ($w_c > 0$) 为水印嵌入容量 WEC 的权重值, w_d ($w_d > 0$) 是数据失真 AoDD 的权重值, 满足 $w_c - w_d = 1$ 。可以根据对水印嵌入容量和数据失真的不同需求动态调整训练参数。

(2) 最佳嵌入属性位选取

本文利用自适应差分进化算法^[15]进行水印最佳嵌入位置的选择, 在提高嵌入容量的同时数据失真也会比较大。因此, 通常选择每一个元组差值最小的两个属性进行嵌入, 但是该算法难以

抵御最小差值攻击。于是, 在嵌入水印之前增加一次属性失真的判断, 将超过属性失真范围的属性位置用最小差值计算的属性位置来代替, 具体步骤如下。

步骤1 初始化基因种群 X 。首先根据式(8)随机产生包含 N 个个体的初始基因种群 $X = \{x_1^0, x_2^0, \dots, x_N^0\}$, x_i^0 基因个体是一个浮点数, 表示当前数据库子组被选择属性列的决策值, 其中, 整数部分表示第一个嵌入水印的属性位, 小数部分表示第二个嵌入水印的属性位。确定交叉算子 CR, 确定最大的进化迭代次数 $\text{EI}_{\max}$, 初始 $\text{EI}=0$ 。

$$x_i^0 = \text{rand}(0,1) \cdot (\text{ub} - \text{lb}) \quad (8)$$

其中, ub 和 lb 分别为 x_i^0 取值的上下界, $\text{rand}(0,1)$ 表示在区间(0,1)内随机取一个数。

步骤2 根据式(9)~式(11)对初始基因种群进行变异处理得到变异后的基因个体数组 V 。

$$V_i^{\text{EI}+1} = x_{\text{best}}^{\text{EI}} + F(x_{i_1}^{\text{EI}} - x_{i_2}^{\text{EI}}) \quad (9)$$

$$V_i^{\text{EI}+1} = x_i^{\text{EI}} + F(x_{i_1}^{\text{EI}} - x_{i_2}^{\text{EI}}) + F(x_{\text{best}}^{\text{EI}} - x_i^{\text{EI}}) \quad (10)$$

$$F' = \begin{cases} 1 - \sqrt[3]{\frac{f(V_i^{\text{EI}})}{f(x_{\text{best}}^{\text{EI}})}}, & \text{Func}_{\text{obj}} = \text{Func}_{\max} \\ 1 - \sqrt[3]{\frac{f(x_{\text{best}}^{\text{EI}})}{f(V_i^{\text{EI}})}}, & \text{Func}_{\text{obj}} = \text{Func}_{\min} \end{cases} \quad (11)$$

步骤3 生成随机值 rand_j , $\text{rand}_j \in [0,1]$, 如果 $\text{rand}_j \leq \text{CR}$ 或者 $j = j_{\text{rand}}$, 则变异个体保留到下一代; 否则当前个体保留到下一代。根据式(12)将生成的变异个体和当前个体进行交叉操作, 生成交叉后的个体数组 U 。

$$U_{ij}^{\text{EI}+1} = \begin{cases} V_{ij}^{\text{EI}+1}, & \text{rand}_j \in [0,1] \leq \text{CR} \text{ 或者 } j = j_{\text{rand}} \\ x_{ij}^{\text{EI}}, & \text{其他} \end{cases} \quad (12)$$

($j=1,2,\dots,N$)

步骤4 分别计算 U 和 X 的适应度函数, 适应度函数越小, 说明越符合收敛目标。

步骤5 根据步骤1重新随机产生一个新的初始种群 X' , 分别计算 X' 和 X 的适应度函数。

步骤6 重复执行步骤2到步骤5, 直到获得最优决策向量或者达到最大的进化迭代次数, 计

算得到的最优值即当前数据库子组的最佳待嵌入水印属性位,与元组主键一起构成水印嵌入位置表 $adet$ 。

步骤 7 提取每个参与水印运算的元组取余后属性差值最小的两个属性,从而记录最小差值的水印嵌入位置表 mdt 。对 $adet$ 进行属性失真判断,如果元组嵌入水印后的值超过了有失真范围,则将其水印嵌入位的下标记为-1,然后把下标为-1的位置用 mdt 的下标来代替,生成最终的嵌入水印的属性位置表 ade_mdt 。

(3) 水印嵌入算法

差分扩展技术^[18]的核心思想是利用数值间的平均值和差值计算将水印嵌入数据载体中。基于这一特性,差分扩展的水印嵌入率很高,但同时如果数据之间的差值过大,刚经过计算后得到的新数据的改变量也很大,数据的失真降低了数据水印的可用性。因此本文对要嵌入水印的数据进行有条件的取余,尽量将数据之间的差值控制在一定范围内,优化差分扩展特性导致的数据失真过大的弊端,增大水印的嵌入容量。

假设要嵌入水印的属性值为 A_1 、 A_2 ,待嵌入的水印比特为 W_i ,改进的差分扩展技术采用取余的方式降低数值间的差值,如式(13)所示。

$$\begin{cases} a_1 = A_1 \bmod Val_{mod} \\ a_2 = A_2 \bmod Val_{mod} \\ div_1 = A_1 / Val_{mod} \\ div_2 = A_2 / Val_{mod} \end{cases} \quad (13)$$

其中, a_1 、 a_2 为取余后的属性值, Val_{mod} 为取余值, div_1 、 div_2 表示对应的除数。

取余后属性值间的均值 avg 和差值 d 计算如式(14)所示。

$$\begin{cases} avg = \left\lfloor \frac{a_1 + a_2}{2} \right\rfloor \\ d = a_1 - a_2 \end{cases} \quad (14)$$

其中, $\lfloor x \rfloor$ 表示向下取整,在水印嵌入阶段,嵌入水印的方式是对差值进行扩展,如式(15)所示。

$$d' = 2d + W_i \quad (15)$$

得到嵌入水印的取余属性值 a'_1 、 a'_2 ,计算如式(16)所示。

$$\begin{cases} a'_1 = avg + \left\lfloor \frac{d' + 1}{2} \right\rfloor \\ a'_2 = avg - \left\lfloor \frac{d'}{2} \right\rfloor \end{cases} \quad (16)$$

取余数和除数相乘加上嵌入水印的取余属性值还原得到最终嵌入水印的属性数值 A'_1 、 A'_2 。计算如式(17)所示。

$$\begin{cases} A'_1 = Val_{mod} \cdot div_1 + a'_1 \\ A'_2 = Val_{mod} \cdot div_2 + a'_2 \end{cases} \quad (17)$$

传统差分扩展技术和改进差分扩展技术对数值的改变量受到原始数据差值的影响,根据图3可以看出,在原始数据差值小于10时,传统的差分扩展的数据改变量普遍小于改进的差分扩展,而在原始数据差值大于10时,改进的差分扩展的数据改变量明显小于传统的差分扩展。因此,选取嵌入水印的属性位置后,先判断该位置上的属性差值 d ,如果 $d < 10$,则选择传统的差分扩展算法,反之选择改进的差分扩展算法。

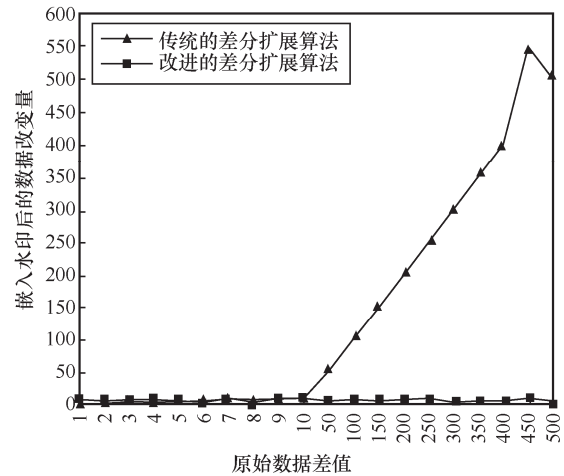


图3 传统的差分扩展和改进的差分扩展算法应嵌入水印后的数据改变量对比

Figure 3 Comparison of the amount of data change between the traditional differential extension algorithm and the improved differential extension algorithm after embedding the watermark

基于自适应差分进化和动态差分扩展的水印嵌入算法如算法1所示。首先对数据库进行聚类分为 K 个子组;其次对所有子组使用自适应差分进化算法初步选择出水印嵌入位,通过有失真分析结合最小差值算法确定最佳的水印嵌入位;最后判断嵌入水印数值间的差值,根据差值的范围动态选择传统的或改进的差分扩展算法,依次对

子组进行水印的嵌入。特别地，每个子组冗余地嵌入同一水印比特位。

算法1 ADE-DDEW 水印嵌入算法

输入 数据库 DB，数据库分组数 K ，嵌入水印属性列数 N ，待嵌入水印序列 W ，取余值 Val_{mod}
输出 嵌有水印的数据库 DB' ，最终的嵌入水印位置表 ade_mdt

```

1)  $\text{ade\_mdt} = []$  //初始化最终的嵌入水印位置表  $\text{ade\_mdt}$ 
2)  $\text{adet} = []$  //初始化根据自适应差分进化计算的水印最佳嵌入属性位置表  $\text{adet}$ 
3)  $\text{mdt} = []$  //初始化根据最小差值计算的水印最佳嵌入属性位置表  $\text{mdt}$ 
4)  $\text{DB}_i \leftarrow \text{kmeans}(\text{DB}, K)$  //利用  $K$ -means 算法进行数据库分组，得到  $K$  个子组
5) For  $i=0$  to  $K-1$  do //遍历每个子组嵌入水印，得到嵌有水印的数据库  $\text{DB}$ 
6)  $\text{adet} \leftarrow \text{ADE}(\text{DB}_i, K, N)$  //自适应差分进化算法计算出最佳属性下标存放在  $\text{adet}$  数组中
7)  $\text{mdt} \leftarrow \text{MD}(\text{DB}, K, N)$  //最小差值函数计算出差值最小的属性下标存放在数组  $\text{mdt}$  中
8) If  $\text{Distortion}(\text{adet}) \notin [A_{\text{miw}}, A_{\text{max}}]$  then //模拟嵌入水印判断是否超过允许有失真范围
9)  $\text{adet}[1][2] = -1$  //如果超过有失真范围，将有失真属性位置下标置为-1
10) End if
11)  $\text{ade\_mdt} \leftarrow$  将  $\text{adet}$  数组中下标为-1 的部分用  $\text{mdt}$  的下标代替
12) For  $j=0$  to  $\text{ade\_mdt}.\text{GetLength}(0)-1$  do //遍历当前子组嵌入水印位置表  $\text{ade\_mdt}$  的嵌入位置
13) For  $k=1$  to  $\text{ade\_mdt}[0].\text{length}-1$  do
14) If  $\text{ade\_mdt}[j][k] \neq -1$  then
15) If  $d < 10$  then...//嵌入水印数值的差值小于 10
16) 利用传统的差分扩展技术嵌入水印  $W_i$ 
17) Else
18) 利用改进的差分扩展技术嵌入水印  $W_i$ 
19) End if

```

20) $\text{DB}_i \leftarrow$ 根据式(17)还原嵌入水印的属性值

21) End if

22) End for

23) End for

24) $\text{DB}' = \bigcup_{i=1}^K \text{DB}'_i$ //各个子数据库合

并成嵌有水印的数据库

25) Return DB' , ade_mdt

26) End for

(4) 水印提取算法

在水印提取阶段，同样通过聚类算法进行分组。对每个子组提取水印比特位，由于是冗余嵌入，利用大数表决的方法确定提取的水印数值，将提取出的水印比特位按照分组顺序组合在一起得到最终的水印序列，同时能恢复出原始的数据库。

水印提取算法首先对嵌入水印的数据库 DB' 进行聚类处理，分成若干个子组，然后找到子组嵌入水印的属性位，依次对每个元组嵌入水印的属性值进行差分扩展逆运算处理，提取出每个子组嵌入的水印集合，对集合进行大数表决选取占比超过 50% 的数作为该子组最终提取的水印比特位，按照分组顺序组合水印比特得到提取的水印序列，提取水印后的属性值恢复成原始数值合并子组得到原始数据库。差分扩展逆运算的具体操作如下。

假设 A'_1 、 A'_2 是嵌入水印的属性数值， a_1 、 a_2 是取余后的属性值，对 A'_1 、 A'_2 进行取余操作如式(18)所示。

$$\begin{cases} \text{arr}'_1 = A'_1 \bmod \text{Val}_{\text{mod}} - a_1 \\ \text{arr}'_2 = A'_2 \bmod \text{Val}_{\text{mod}} - a_2 \end{cases} \quad (18)$$

其中， arr'_1 、 arr'_2 是提取过程中的数值属性取余值，对 arr'_1 、 arr'_2 进行均值 avg' 和差值 d' 计算，如式(19)所示。

$$\begin{cases} \text{avg}' = \left\lfloor \frac{\text{arr}'_1 + \text{arr}'_2}{2} \right\rfloor \\ d' = \text{arr}'_1 - \text{arr}'_2 \end{cases} \quad (19)$$

通过 d' 来进行水印比特的提取，如式(20)所示。

$$b = d' - 2 \left\lfloor \frac{d'}{2} \right\rfloor \quad (20)$$

随后利用 avg' 、 d' 即可恢复取余后的属性值

arr_1 、 arr_2 如式(21)所示。

$$\begin{cases} \text{arr}_1 = \text{avg}' + \left\lfloor \frac{d' / 2 + 1}{2} \right\rfloor \\ \text{arr}_2 = \text{avg}' - \left\lfloor \frac{d' / 2}{2} \right\rfloor \end{cases} \quad (21)$$

利用除数和取余值进行计算恢复原始属性值 A_1 、 A_2 如式(22)所示。

$$\begin{cases} A_1 = \text{Val}_{\text{mod}} \cdot \text{div}_1 \\ A_2 = \text{Val}_{\text{mod}} \cdot \text{div}_2 \end{cases} \quad (22)$$

ADE-DDEW 水印提取算法在利用改进的差分扩展算法进行水印提取时,通过嵌入水印时的除数和取余值恢复数据,数据可以复原,即使数据库遭到一定破坏,提取的水印可能不完整,但数据库可以复原如初,有效保障了数据库的可用性。针对提取出的水印序列,根据数据压缩恢复一维水印序列,根据 SVD 逆运算还原出低频图片,最后通过 Haar 小波逆变换还原出 QR 码。根据扫描出的 QR 码信息,可以确定泄密者的身份,识别攻击者。特别地,即使还原的 QR 码有失真,只要在一定范围内也可以扫描出身份信息,提高了水印的鲁棒性,如算法 2 所示。

算法 2 ADE-DDEW 水印提取算法

输入 数据库 DB' , 嵌入水印位置表 ade_mat , 数据库分组数 K , 取余值 Valmod

输出 恢复的原始数据库 DB , 提取的数字水印序列 W , 还原出的 QR 码图像

- 1) $\text{DB}_i \leftarrow \text{kmeans}(\text{DB}', K)$ //利用 K -means 算法进行数据库分组,得到 K 个子组
- 2) For $i=0$ to $K-1$ do //迭代每个子组
- 3) $v_0=0, v_1=0$ //初始化记录提取的 0、1 个数的变量,用于大数表决
- 4) For $j=0$ to $\text{ade_mdt}.\text{GetLength}(0)-1$ do //迭代每个子组的所有元组
- 5) $A'_1 = \text{ade_mdt}[j][0]$, $A'_2 = \text{ade_mdt}[j][1]$ //根据 ade_mdt 选择出嵌入水印的属性
- 6) 根据式(18)对 A'_1 、 A'_2 进行取余操作
- 7) 根据式(20)提取水印的比特值 W_i , 根据式(21)和式(22)恢复属性数值 A_1 、 A_2
- 8) If $W_i == 0$ then

- 9) v_0++ //统计提取的水印比特值为 0 的个数
- 10) Else
- 11) v_1++ //统计提取的水印比特值为 1 的个数
- 12) End if
- 13) End for
- 14) If $v_0 > v_1$ then //大数表决
- 15) $W_i = 0$ //如果提取的水印 0 的个数大于 1 的个数,则该子组的水印比特为 0
- 16) Else
- 17) $W_i = 1$ //否则该子组的水印比特为 1
- 18) End if
- 19) $W.\text{append}(W_i)$ //依次组合每个子组提取的水印比特值
- 20) $\widetilde{\text{QR}} \leftarrow \text{SVD}(\widetilde{W}, U, D)$ // $\widetilde{W}$ 和左右奇异值矩阵相乘得到低频图像矩阵 $\widetilde{\text{QR}}$
- 21) $\text{QR} \leftarrow \text{Har}(\widetilde{\text{QR}})$ //通过 Haar 小波逆变换还原出 QR 码图像
- 22) $\text{DB} = \bigcup_{i=1}^K \text{DB}_i$ //各个子数据库合并成原始数据库
- 23) End for
- 24) Return DB, W, QR

2 实验与结果分析

2.1 实验指标

本文的仿真实验采用 UCI 机器学习资源库提供的 Appliances energy prediction Data Set 作为数据样本,该数据库包含 19 735 个元组和 29 个属性。由于 ADE-DDEW 水印算法是对整数型的数值属性进行计算,所以对于浮点型属性值进行预处理,将其扩大成整数型数据然后通过差分扩展嵌入水印,最后按比例还原成浮点型。将 ADE-DDEW 算法与已有的 GADEW^[10]、GAHSW^[8]、RF-GADEW^[14]和 RF-GAHSW^[14]等可逆数据库水印算法进行对比实验分析。

本文通过对比原始水印比特位和提取的水印比特位的方法来衡量数据库水印算法的鲁棒性,即水印的误码率 (BER, bit error rate)^[8]。鲁棒性评价指标 BER 的计算如式(23)所示。

$$\text{BER} = \frac{1}{K} \sum_{i=1}^K W_i \oplus W'_i \quad (23)$$

其中, K 为数据库的分组数, 即水印序列的长度, W_i 和 W'_i 分别是嵌入水印和提取水印的第 i 位比特值。由式(23)可知, 算法的鲁棒性和 BER 成反比, 较低的误码率值意味着较高的水印鲁棒性。

对于数据库水印算法的嵌入容量, 通过水印嵌入容量 (WEC) 来衡量, 即计算嵌入水印的数据库元组占数据库元组的比例。对于数据失真, 即嵌入水印后造成数据库数据的改变量。本文通过水印嵌入前后属性的平均值 (Mean)、标准差 (Std) 的变化以及平均绝对误差 (MAE) 来量化其统计失真, MAE 的计算如式(24)所示。

$$\text{MAE} = \frac{\sum_{i=1}^n |A_i - A'_i|}{n} \quad (24)$$

其中, n 为数据库元组的个数, A_i 和 A'_i 分别为嵌入水印前后对应的属性值。

2.2 相关参数分析

在将水印嵌入数据库之前, 需要对 QR 码和数据库进行预处理, 使得嵌入的水印序列包含更多信息, 提高嵌入容量。为了在相同长度下得到更多信息, 经常采用数据压缩技术, 这其中 Haar 小波变换效果最好, 嵌入数字 QR 码和嵌入文本 QR 码的小波变换对比分别如表 1 和表 2 所示。

压缩率和绝对最大差异是衡量小波变换压缩效果的指标。压缩率越高, 表示数据压缩的能力越好, 信息密度越高; 绝对最大差异越大, 表示数据信息的多样性越好, 过滤信号的能力越强。本文对不同像素的 QR 码进行了小波变换实验, 从表 1 和表 2 可以看出, Haar 小波变换的压缩率和绝对最大差异综合强于另外 3 种小波变换 (Daubechies、Coiflet、Symlet) 算法, 这也是文本选用 Haar 小波变换压缩 QR 码的理由。

改进差分进化算法中的取余值 Val_{mod} 决定了数据的改变量, 取余值过小, 数据碰撞频繁, 嵌入难度大; 取余值过大, 数据失真增大, 数据库的可用性降低。本文对 5~30 的取余值进行了实验, 计算其平均数值改变量, 取余值 Val_{mod} 与数值改变量的关系如图 4 所示, 可以看出取余值 $\text{Val}_{\text{mod}}=7$ 时, 数据的平均变化量最少。

ADE-DDEW 水印算法采用自适应差分进化算法计算最佳属性位, 不同的种群规模对算法最优解的计算有一定的影响, 种群规模过大导致算法的收敛速度下降, 搜索时间增加; 种群规模过小容易过早收敛, 全局搜索能力差, 影响算法的鲁棒性。对区间[10,120]的种群规模 P 进行测试, 不同种群规模下适应度值对比如图 5 所示, 结果显示, 当种群规模为 90 时, 目标函数适应度值最小。

表 1 嵌入数字 QR 码的小波变换对比
Table 1 Comparison of wavelet transform with embedded digital QR code

小波变换	QR 码像素为 64×64		QR 码像素为 128×128		QR 码像素为 256×256	
	压缩率	绝对最大差异	压缩率	绝对最大差异	压缩率	绝对最大差异
Haar	70.7 001 953	5.4 001 247	78.3 752 441	7.3 896 444	93.3 868 408	9.094 947
Daubechies	68.8 720 703	1.0 516 032	74.2 797 851	1.2 221 335	82.588 195	1.4 687 962
Coiflet	71.508 789	1.4 861 683	75.8 422 851	7.6 738 615	84.6 649 169	1.8 301 996
Symlet	69.2 626 953	1.0 461 462	74.1 210 937	9.0 750 518	84.1 064 453	2.1 753 364

表 2 嵌入文本 QR 码的小波变换对比
Table 2 Comparison of wavelet transform with embedded text QR code

小波变换	QR 码像素为 64×64		QR 码像素为 128×128		QR 码像素为 256×256	
	压缩率	绝对最大差异	压缩率	绝对最大差异	压缩率	绝对最大差异
Haar	69.5 751 953	5.9 685 589	83.203 125	7.3 896 444	89.979 553	7.6 738 615
Daubechies	67.8 955 071	9.3 791 641	77.1 911 621	1.1 937 117	81.054 687	1.4 164 767
Coiflet	68.2 617 187	1.3 672 206	79.461 669	1.4 278 214	81.561 279	1.938 388
Symlet	68.2 617 187	5.6 616 581	77.746 582	7.7 305 807	81.381 225	2.1 638 108

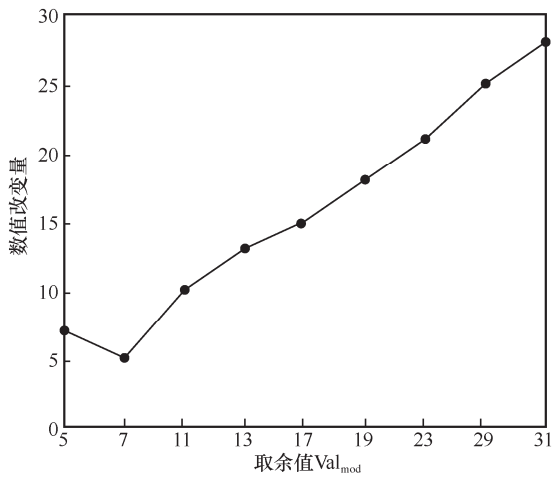


图4 取余值 Val_{mod} 与数值改变量的关系
Figure 4 The residual value Val_{mod} is plotted against the amount of change in value

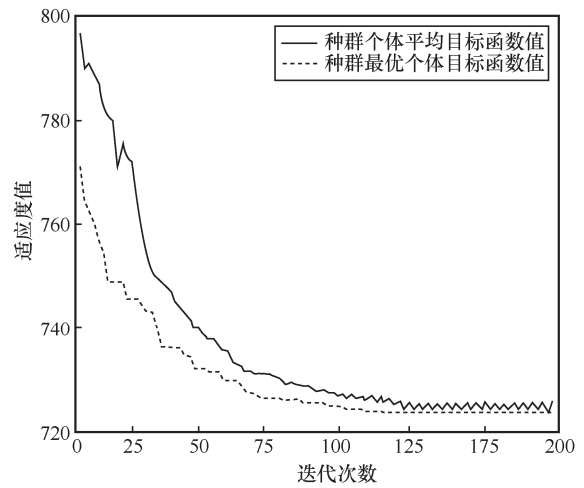


图6 适应度值和迭代次数的关系
Figure 6 Plot of the relationship between fitness value and number of iterations

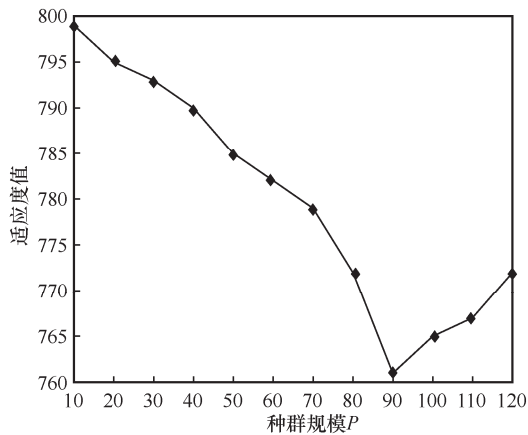


图5 不同种群规模下适应度值对比
Figure 5 Comparison of the adaptation of different population sizes

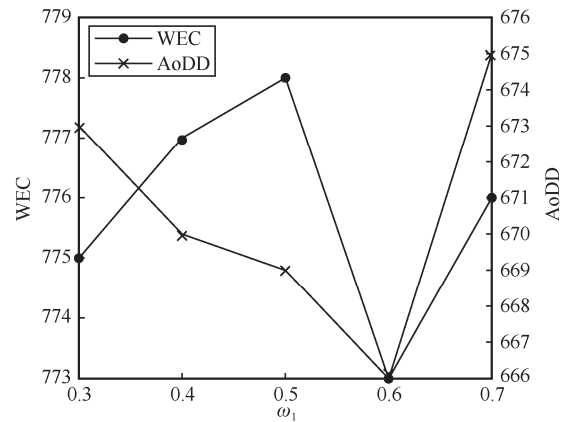


图7 权重系数与目标函数关系
Figure 7 Graph of weight coefficients versus objective function

除此之外,本文评估了迭代次数对于自适应差分进化算法的影响,分别进行了25、50、75、125、150、175和200次迭代。每迭代一次,ADE-DDEW算法给数据库随机产生一个新的初始种群参与族外竞争,避免差分进化陷入局部最优,适应度值和迭代次数的关系如图6所示。可以看出,当迭代次数为200时,ADE-DDEW算法达到最低的适应度值。

实验还对目标函数中的权重系数 ω_1 和 ω_2 进行了评估,因为 $\omega_2=1-\omega_1$,因此本文只对 ω_1 进行了实验,选取水印嵌入容量(WEC)和数据失真量(AoDD)作为评价指标,权重系数与目标函数关系如图7所示。从图中可以看出,当 $\omega_1=0.6$ 时实验结果最好。

不同的嵌入属性列数可能会导致容量和数据失真不同,本文对10~29列的情况分别做实验,具体结果如图8所示。

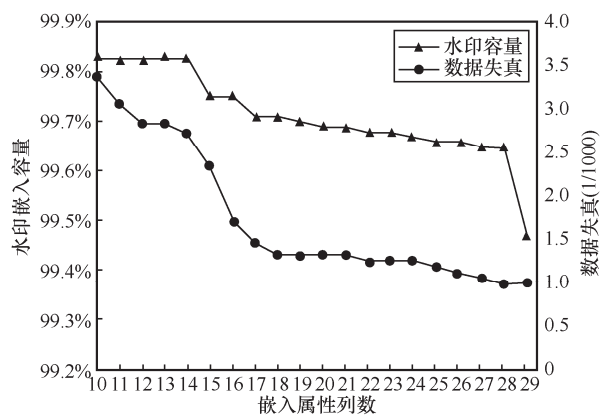


图8 不同嵌入属性列数下水印嵌入容量和数据失真对比
Figure 8 Comparison of watermark embedding capacity and data distortion with different number of embedding columns

实验结果表明,选取 28 列属性嵌入水印,其嵌入容量较高、数据失真最小。因此,后面的实验都将选取 28 列属性进行水印的嵌入。

综上所述, ADE-DDEW 水印嵌入算法设置的取余值 $\text{Val}_{\text{mod}}=7$, 目标函数中的 $\omega_1=0.6$, $\omega_2=0.4$, 种群规模 P 的大小为 90, 迭代次数为 200, 选取属性列数为 28。

2.3 实验过程

(1) 嵌入容量和数据失真对比实验

将本文提出的 ADE-DDEW 算法与已有的 GADEW、RF-GADEW、GAHSW 和 RF-GAHSW 算法进行水印嵌入容量和数据失真的对比。对于数据库水印算法来说,嵌入容量越大,数据失真越大。各水印算法嵌入容量和数据失真对比如图 9 所示。

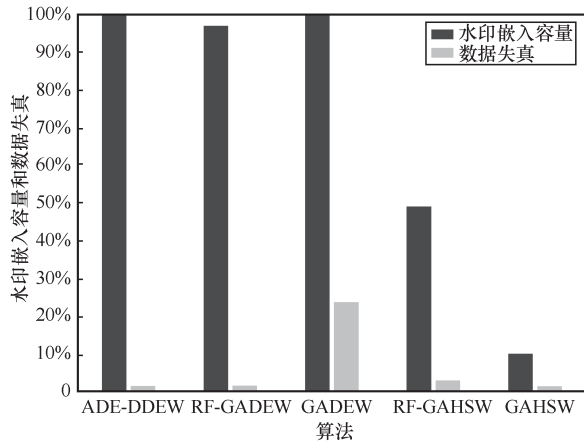


图9 各水印算法嵌入容量和数据失真对比

Figure 9 Comparison of embedding capacity and data distortion by watermarking algorithm

从图 9 中可以看出,本文提出的 ADE-DDEW 算法在嵌入容量高的算法中数据失真最低,在数据失真低的算法中嵌入容量最高,有效缓解了水印嵌入容量和数据失真之间存在的矛盾。ADE-DDEW 算法对数据进行了取余操作,限制了数据的变化范围,同时浮点型化为整型计算有效降低了数值改变量,因此水印不会造成过大的数据失真,采用差分扩展算法嵌入水印同时结合最小差值和差分进化算法确定水印嵌入位置提高了水印的嵌入容量。

为了观察嵌入水印后数据的变化量,本文通过对比平均值、标准差和 MAE 的变化来评估统计数据失真。选取嵌入水印前后的数据库属性,取 10 次模拟实验的平均值作为结果。水印嵌入前后不同数据库水印算法的平均值和标准差统计结果如表 3 所示,其中加粗的数据表示最优的结果。为了更直观地看到平均值和方差的变化,本文给出了每个属性的平均数之差 D_M 和标准差之差 D_S , 水印嵌入前后不同数据库水印算法的平均值、标准差变化和平均绝对误差统计结果如表 4 所示,不同算法的 MAE 值显示在最后一行。 D_M 和 D_S 的计算如式(25)和式(26)所示。

$$D_M = |\text{Mean} - \text{Mean}'| \quad (25)$$

$$D_S = |\text{Std} - \text{Std}'| \quad (26)$$

其中, Mean 和 Std 分别代表原始数据库的平均值和标准偏差值, Mean' 和 Std' 分别代表嵌入水印后

表3 水印嵌入前后不同数据库水印算法的平均值和标准差统计结果

Table 3 Mean and standard deviation statistics of different database watermarking algorithms before and after watermark embedding

属性	原始数据库		ADE-DDEW 算法		GADEW 算法		RF-GADEW 算法		GAHSW 算法		RF-GAHSW 算法	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
T9	19.483	2.015	19.476	2.009	19.082	2.382	19.575	2.133	19.483	2.015	19.453	2.035
RH_9	41.548	4.151	41.541	4.143	41.383	4.601	41.476	4.285	41.572	4.139	41.563	4.148
T_out	7.4	5.317	7.389	5.308	7.201	5.528	7.523	5.203	7.443	5.348	7.432	5.329
Press_mm_hg	79.769	7.399	79.757	7.405	79.829	7.021	79.892	7.256	79.737	7.371	79.748	7.385
RH_out	755.523	14.901	755.518	14.901	755.401	14.710	755.437	14.834	755.556	14.850	755.549	14.924
Windspeed	4.041	2.451	4.052	2.459	4.284	2.649	4.184	2.392	4.086	2.411	4.067	2.428
Visibility	38.342	11.794	38.35	11.789	38.742	11.620	38.492	11.630	38.393	11.756	38.361	11.775
Tdewpoint	3.753	4.195	3.747	4.204	3.932	4.348	3.629	4.275	3.702	4.264	3.737	4.218
rv1	24.986	14.496	24.972	14.511	24.847	14.730	24.816	14.583	24.932	14.558	24.971	14.526

表 4 水印嵌入前后不同数据库水印算法的平均值、标准差变化和平均绝对误差统计结果
Table 4 Mean, standard deviation change and mean absolute error statistics of different database watermarking algorithms before and after watermark embedding

属性	ADE-DDEW 算法		GADEW 算法		RF-GADEW 算法		GAHSW 算法		RF-GAHSW 算法	
	D_M	D_S	D_M	D_S	D_M	D_S	D_M	D_S	D_M	D_S
T9	0.001	0.002	0.023	0.063	0.002	0.192	0	0	0.003	0.045
RH_9	0.001	0.003	0.019	0.320	0.187	0.073	0.006	0.072	0.084	0.065
T_out	0.002	0.001	0.380	0.754	0.003	0.018	0.038	0.047	0.026	0.194
Press_mm_hg	0.002	0.006	0.049	0.128	0.218	0.007	0.014	0.005	0	0
RH_out	0.003	0.002	0.020	0.870	0.010	0.501	0.106	0.023	0.008	0.073
Windspeed	0.001	0.001	0.076	0.018	0.009	0.129	0.080	0.007	0.017	0.072
Visibility	0.002	0.001	0.890	0.342	0.227	0.284	0.004	0.039	0.080	0.160
Tdewpoint	0.004	0.001	0.017	0.050	0.001	0.547	0.072	0.216	0.021	0.015
rv1	0.003	0.002	0.430	0.081	0.039	0.028	0.193	0.074	0.047	0.007
MAE	0.019		2.914		1.315		0.852		0.516	

数据库的平均值和标准偏差值。

由表 3 和表 4 可知, ADE-DDEW 算法的各个属性在平均值和标准差方面的变化非常小, 其造成的最大变化是 0.015, 这与 GADEW、RF-GADEW、GAHSW 和 RF-GAHSW 算法相比可以忽略不计。另外, 表中有一些变化值为 0, 表示该属性列没有嵌入水印。一般来说, 水印算法对数据质量的影响越小越好, 对于量化属性分布有失真的 MAE 来说也是如此。表中 GADEW、RF-GADEW、GAHSW 和 RF-GAHSW 算法的 MAE 值并不理想, 分别为 2.914、1.315、0.852 和 0.516, 这意味着嵌入水印后会造巨大的数据失真, 影响数据库的可用性, 而 ADE-DDEW 算法的 MAE 值为 0.019, 远远低于其他算法。实验结果验证了所提 ADE-DDEW 算法的有效性。综上所述, ADE-DDEW 算法在统计失真方面的表现明显优于其他算法, 保障了数据库的可用性。

(2) 鲁棒性实验

本文通过检测水印的误码率 (水印检测失败的比率) 来判断在多种元组攻击下各数据库水印算法的鲁棒性, 其中误码率越低, 说明水印鲁棒性越好, 当误码率接近零时, 水印被正确地数据库检测出来。实验分析和对比了本文提出的 ADE-DDEW 算法与 GADEW、RF-GADEW、GAHSW 和 RF-GAHSW 算法在删除攻击、修改攻击和插入攻击下水印的误码率, 分别完成 0、

10%、20%、30%至 90%的攻击比例, 取误码率的平均值作为最终结果。实验结果如图 10~图 12 所示。

1) 元组删除攻击

攻击者会随机删除元组以破坏水印, 被删除的元组中可能嵌有水印, 如果删除过多, 会导致无法提取出正确的水印, 影响版权的保护。删除攻击下各算法提取水印的误码率对比如图 10 所示。

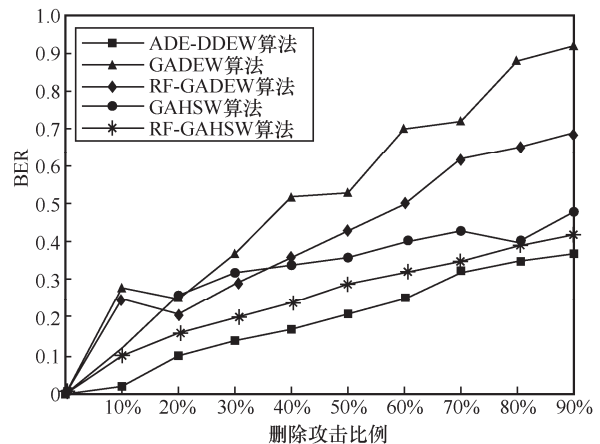


图 10 删除攻击下各算法提取水印的误码率对比
Figure 10 Comparison of BER of watermark extraction by algorithms under removal attack

从图 10 可以看出, 当数据库受到严重删除攻击时, 如数据库中被删除的元组达到 90%, GADEW、RF-GADEW、GAHSW 和 RF-GAHSW 算法提取水印的 BER 值分别为 0.93、0.74、0.47

和 0.42，而 ADE-DDEW 算法的 BER 值为 0.37。如果删除的元组中含有大量的水印信息，会影响到水印的完整提取。因此，删除攻击对数据库水印的影响最为严重。实验还表明，随着删除元组数量的增加，水印提取时的误码率随之增加，相较于其他方法，ADE-DDEW 算法的误码率曲线增长较为平缓且最大的误码率不超过 0.4，在删除攻击下有更好的鲁棒性。

2) 元组修改攻击

攻击者会随机修改元组中的任意属性值以破坏水印，被修改的属性值中可能嵌有水印，如果修改过多，会导致无法提取出正确的水印，同时还还原数据库过程也会受到影响。修改攻击比例下各算法提取水印的误码率对比如图 11 所示。

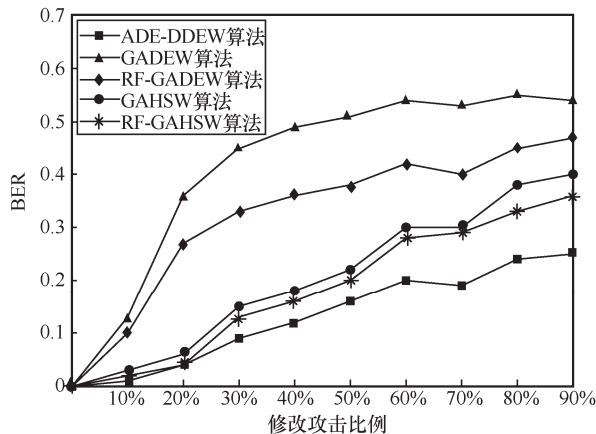


图 11 修改攻击下各算法提取水印的误码率对比
Figure 11 Comparison of BER of watermark extraction by each algorithm under modified attack

从图 11 可以看出，修改攻击的强度和水印的误码率成正比，ADE-DDEW 算法的整体误码率明显低于其他水印算法，即使修改了 90% 的元组属性值，ADE-DDEW 算法的误码率也只有 0.25，也能恢复大部分的水印信息和原始数据，是高度稳健的水印算法，有利于溯源攻击者。

3) 元组插入攻击

元组插入攻击是随机创建元组并插入数据库以破坏水印的一种攻击。元组插入攻击下提取水印的误码率对比如图 12 所示。

实验结果表明，对于利用差分扩展进行水印嵌入的数据库水印算法来说，向数据库添加元组这种攻击方式不会对水印信息完整正确地提取造成影响，也就是说，无论插入多少元组，在插入

攻击下的误码率始终为 0。因此，ADE-DDEW 算法对插入攻击具有极强的鲁棒性。

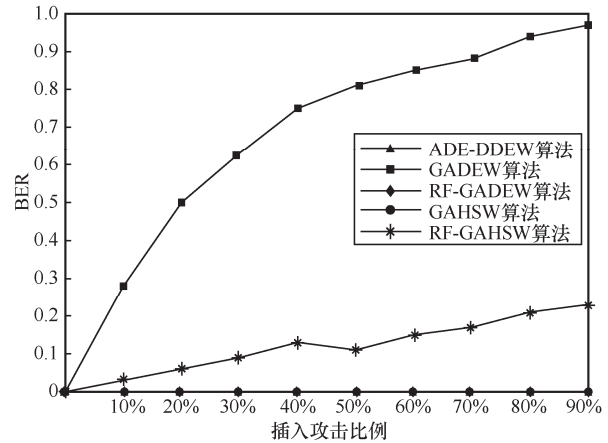
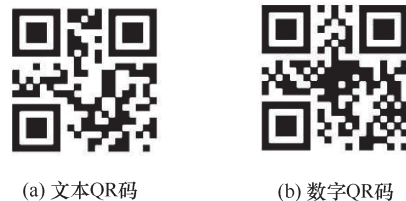


图 12 元组插入攻击下提取水印的误码率对比
Figure 12 Comparison of BER of extracted watermark under tuple insertion attack

综上所述，ADE-DDEW 算法在删除攻击、修改攻击和插入攻击下水印误码率增长缓慢，特别是对于插入攻击，可以完全提取出正确的水印。可见 ADE-DDEW 算法可以抵御常见的数据库攻击，具有良好的鲁棒性。

(3) 溯源效果展示

ADE-DDEW 算法嵌入的水印是经过压缩的 QR 码，其中文本 QR 码带有“njupt”的身份信息，数字 QR 码带有“20211216”的身份信息。提取出的水印序列通过一系列的逆运算还原出 QR 码，扫描 QR 码就可以得到唯一标识攻击者的身份信息，以达到追踪溯源的目的。本文提取攻击实验后的水印序列，将其还原成对应的 QR 码与嵌入的 QR 码进行对比，检测算法的追踪溯源能力。待嵌入 QR 码图像如图 13 所示。



(a) 文本 QR 码 (b) 数字 QR 码

图 13 待嵌入 QR 码图像
Figure 13 QR code image to be embedded

各攻击下提取的 QR 码图像如表 5 所示。QR 码有极强的容错率，即使提取的水印存在一定的无码率，还原的 QR 码也能够扫描出身份信息。

由上述图表可知，即使在攻击比例高（90%）的情况下，扫描 QR 码仍然可以准确无误地扫描出“njupt”和“20211216”这两个身份信息，能够准确地追踪到攻击者。

表 5 各攻击下提取的 QR 码图像
Table5 QR code images extracted under each attack

攻击比例	删除攻击		修改攻击		插入攻击	
	文本 QR 码	数字 QR 码	文本 QR 码	数字 QR 码	文本 QR 码	数字 QR 码
10%						
90%						

（4）数据库还原实验

ADE-DDEW 算法在提取水印的过程中可以恢复原始的数据值。由上述实验可知，ADE-DDEW 算法可以 100%抵御插入攻击，因此不涉及插入攻击。本文对数据库进行修改元组值、删除元组和删除属性攻击，将提取水印前后不一样的数据记为错误数据，记数值改变率 = $\frac{\text{错误数据的个数}}{\text{当前嵌入水印数据的个数}}$ ，作为衡量算法还原性的指标。实验分析不同攻击比例下，ADE-DDEW 算法修改元组、删除元组和删除属性攻击下数值改变率，结果如图 14 所示。

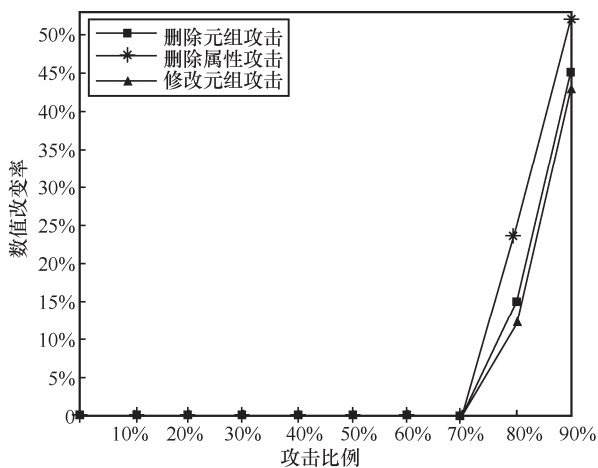


图 14 不同攻击比例下 ADE-DDEW 算法的数据改变情况
Figure 14 Data alteration of ADE-DDEW algorithm for different attack ratios

由图 14 可知，即使大部分数据被攻击，ADE-DDEW 算法也可以恢复原始数据，即使大

量数据被破坏，也可以恢复将近一半的数据，说明 ADE-DDEW 算法具有良好的还原性。

3 结束语

考虑到提高水印嵌入容量和减小数据失真存在的固有矛盾问题，本文提出了一种基于动态差分扩展的强鲁棒数据库水印算法。用 QR 码作为水印载体，利用 Haar 小波变换进行数据压缩并且通过 SVD 提取 QR 码的特征值，作为待嵌入的水印序列，增大了相同水印序列长度下水印包含的信息量。同时过滤多种信号攻击，提高图像抵御攻击的能力。该算法结合自适应差分进化算法和最小差值算法选择最佳嵌入属性位，通过动态地选取传统差分扩展算法或改进的差分扩展技术进行水印的嵌入，以缓解传统差分扩展算法在嵌入水印时计算效率低、数据失真大、鲁棒性差的问题，提高水印嵌入容量的同时减少数据失真。本文通过实验验证了数据库水印算法的可行性，该算法在一定的数据失真范围内能有效地提高水印的嵌入容量，在数据库遭受攻击时表现出良好的鲁棒性，能够有效地溯源泄密者，且与现有的算法对比优势明显。

参考文献：

- [1] HOLT L, MAUFE B G, WIENER A. Encoded marking of a recording signal[P]. U L Patent GB2196167A, 1988.
- [2] HIMANSHU A, FAROOQ H. Development of payload capacity enhanced robust video watermarking scheme based on symmetry of circle using lifting wavelet transform and SURF[J]. Journal of Information Security and Applications, 2021, 59.
- [3] SHOHIDULISLAM M, NAQVI N, ABBASI A T, et al. Robust dual domain twofold encrypted image-in-audio watermarking based on SVD[J]. Circuits, Systems, and Signal Processing, 2021 (prepublish).
- [4] 赵春雨. 提高嵌入容量的水印算法研究[D]. 郑州: 郑州大学, 2011. ZHAO C Y. Research on algorithm for improving the capacity of digital watermarking[D]. Zhengzhou: Zhengzhou University, 2011.
- [5] ZHANG Y, YANG B, NIU X M. Reversible watermarking for relational database authentication[J]. Journal of Computers, 2006, 17(2): 59-66.
- [6] HU D, ZHAO D, ZHENG S. A new robust approach for reversible database watermarking with distortion control[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(6): 1024-1037.
- [7] GUPTA G, PIEPRZYK J. Reversible and blind database watermarking using difference expansion[J]. International Journal of Dig-

- ital Crime and Forensics, 2009, 1(2): 42-54.
- [8] JAWAD K, KHAN A. Genetic algorithm and difference expansion based reversible watermarking for relational databases [J]. Journal of Systems and Software, 2013, 86(11): 2742-2753.
- [9] 邱升红. 面向关系数据库的可逆水印方法研究[D]. 西安: 西安电子科技大学, 2018.
- QIU S H. Research on reversible watermarking methods for relational databases[D]. Xi'an: Xi'an University of Electronic Science and Technology, 2018.
- [10] 宋岩, 沈泉江, 杨洪山. 基于布谷鸟算法的可逆数据库水印方案[J]. 计算机应用与软件, 2021, 38(12): 304-313.
- SONG Y, SHEN Q H, YANG H S. A reversible database watermarking scheme based on cuckoo algorithm[J]. Computer Application and Software, 2021, 38(12): 304-313.
- [11] 孔嘉琪, 王利明, 葛晓雪. 基于模拟退火和粒子群混合改进算法的数据库水印技术[J]. 信息安全, 2022, 22(5): 37-45.
- KONG J Q, WANG L M, GE X X. Database watermarking technique based on simulated annealing and particle swarm hybrid improvement algorithm[J]. Information Network Security, 2022, 22(5): 37-45.
- [12] SHI Y, LI X, ZHANG X, et al. Reversible data hiding: advances in the past two decades[J]. IEEE Access, 2016, 4(10): 3210-3237.
- [13] GE C, SUN J, SUN Y, et al. Reversible database watermarking based on random forest and genetic algorithm[C]//2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). 2020.
- [14] 陈青, 夏兰婷, 卜莹. 基于两级奇异值分解的鲁棒水印算法[J]. 应用科学学报, 2020, 38(6): 966-975.
- CHEN Q, XIA L T, BU Y. Robust watermarking algorithm based on two-level singular value decomposition[J]. Journal of Applied Sciences, 2020, 38(6): 966-975.
- [15] 关虎, 张桂煊, 张树武, 等. 基于二维码的鲁棒图像水印技术及应用研究[J]. 有线电视技术, 2018(10): 20-26.
- GUAN H, ZHANG G X, ZHANG S W, et al. Research on robust image watermarking technology and application based on QR code[J]. Cable TV Technology, 2018(10): 20-26.
- [16] KLEMA V, LAUB A. The singular value decomposition: its computation and some applications[J]. Automatic Control IEEE Transactions on, 1980, 25(2): 164-176.
- [17] STORNR P K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces[R]. 1995.
- [18] TIAN J. Reversible data embedding using a difference expansion[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2003, 13(8): 890-896.

[作者简介]



汪天琦（1997-），女，安徽合肥人，南京邮电大学硕士生，主要研究方向为数字水印技术。



张迎周（1978-），男，安徽庐江人，博士，南京邮电大学教授，主要研究方向为软件与信息安全。



邱云龙（1996-），男，安徽滁州人，博士，南京邮电大学硕士生，主要研究方向为数字水印技术。



李鼎文（1998-），男，江苏南京人，南京邮电大学硕士生，主要研究方向为数字水印技术。



朱林林（1997-），女，山东德州人，南京邮电大学硕士生，主要研究方向为数字水印技术。