基于混沌系统的DCT域加密数字水印嵌入算法

作者姓名 孙恒康

学校导师姓名、职称 李凤华 教授

企业导师姓名、职称 郭云川 高工

申请学位类别 电子信息硕士

学校代码 10701　　学　号 _____
分 类 号 TP309　　密　级 公开

西安电子科技大学

硕士学位论文

基于混沌系统的DCT域加密数字水印嵌入算法

作者姓名：孙恒康

领　　域：网络与信息安全

学位类别：电子信息硕士

学校导师姓名、职称：李凤华 教授

企业导师姓名、职称：郭云川 高工

学　　院：广州研究院

提交日期：2025 年 3 月

DCT-domain encryption digital watermarking embedding algorithm based on chaotic systems

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Electronic Information

By

Zhang San

Supervisor: Li Si Title: Professor

Supervisor: Wang Wu Title: Senior Engineer

March 2024

摘要

随着信息技术的发展，人们在网络上的产生大量的图片信息，直接存储在本地安全性不高，数据有被伪造和被盗用的风险，导致图片版权信息产生争议。如果存证对象不存储在本地而是存储在一个安全的第三方存证中心，虽然安全性增加，但是会增加存证中心的存储压力。一般的做法是向存证系统存证该图片的哈希值和元数据信息（图片名称，拍摄地点，拍摄设备等）。这样即对图片做了存证又避免了大量的数据的存储。

本文在第三方存证方案的基础上，增加了将版权信息以加密的数字水印方式嵌入到本地图片的方案：利用户上传的图片哈希值、元数据以及自定义密钥，生成和存证图片绑定的数字水印，以及将该数字水印嵌入到图片中的嵌入算法。实现侵权发生时可以利用本地图片中的数字水印和第三方存证系统共同验证的双保险存证效果。具体的工作如下：

（1）设计了结合混沌系统和二维码的数字水印的生成和加密方案：该方案将 QR 码和混沌系统结合，将用户上传的图片哈希值和元数据信息利用 SHA256 算法生成唯一的散列值 S。并将散列值 S 生成为 QR 码。然后结合 logistic-tent 系统和洗牌算法将 QR 码完全置乱，隐藏 QR 码的像素的位置信息。随后利用 Chen 混沌系统和 DNA 编码扩散算法将置乱后的 QR 码进行扩散操作，消除 QR 码的黑白像素数量信息。经过以上操作得到的完全随机无序的加密二值数字水印。最后对数字水印的生成和加密方案进行实验，论证本方案的算法可逆性，密钥敏感性，在裁剪、椒盐噪声等攻击下的加密鲁棒性。

（2）结合了人类视觉系统（HVS）和混沌系统的 DCT 域二值图像嵌入算法。具体做法是：将原始图片进行分块，依据人类视觉系统对图片边缘复杂度、纹理复杂度、亮度的不同敏感性，给每块图像打分并排序，分数越高的图像块越适合做水印的嵌入。随后将加密的二值数字水印序列化成 0，1 比特序列，并结合用户提供的密钥和 logistic-tent 混沌系统打乱每个比特嵌入的位置信息。在嵌入过程中结合刚才打分情况使用 DCT 域的数字隐藏算法，越适合嵌入信息的图像块嵌入系数越大，修改幅度越大。以此来平衡数字水印的鲁棒性和不可感知性。最后进行实验，验证算法可逆性，分析在不同嵌入强度下对鲁棒性和不可感知性的影响。

关 键 词：数字水印，混沌系统，HVS，离散余弦变换

ABSTRACT

With the development of information technology, a vast amount of image data is generated online. Direct storage of these images on local systems is insecure, as the data is vulnerable to forgery and theft, leading to disputes over image copyright. If the evidentiary objects are not stored locally but instead in a secure third-party evidence preservation center, although security is enhanced, the storage burden on the preservation center increases. A common approach is to record the image's hash value and metadata (such as image name, shooting location, shooting device, etc.) in the preservation system, which both evidences the image and avoids the need for storing large volumes of data.

This paper builds upon the third-party evidence preservation solution by introducing a method to embed evidentiary information into local images in the form of digital watermarks. Specifically, the proposed scheme utilizes the image's hash value, metadata, and a user-defined key to generate a digital watermark bound to the evidentiary image, as well as an embedding algorithm to incorporate the watermark into the original image. The watermark is then embedded according to this scheme, thereby achieving a dual-layer evidentiary effect whereby infringement can be verified using either the local digital watermark or the third-party preservation system. The specific work is as follows:

(1) First, a chaotic digital watermark generation and encryption scheme is designed. In this paper, QR codes are combined with chaotic systems; the hash value of the user-uploaded image along with its metadata is processed using the SHA256 algorithm to generate a unique hash S, which is then converted into a QR code. Subsequently, by integrating the logistic-tent system with a shuffling algorithm, the QR code is completely scrambled to obscure the positional information of its pixels. Thereafter, the scrambled QR code is diffused using the Chen chaotic system combined with a DNA encoding diffusion algorithm to eliminate quantitative information regarding the black and white pixels in the QR code. The result is a completely random, disordered, and encrypted binary digital watermark. Finally, experiments on the watermark generation and encryption scheme are conducted to demonstrate the algorithm's reversibility, key sensitivity, and robustness against attacks such as cropping and salt-and-pepper noise.

(2) Second, a DCT-domain binary image embedding algorithm that integrates the Human Visual System (HVS) and chaotic systems is proposed. Specifically, the original image is partitioned into blocks, and each block is scored and ranked based on the HVS's varying sensitivity to edge complexity, texture complexity, and brightness; blocks with higher scores are deemed more suitable for watermark embedding. Subsequently, the binary digital watermark is serialized into a bit sequence of 0s and 1s, and in conjunction with the user-provided key and the logistic-tent chaotic system, each bit is randomly assigned to a corresponding image block. During the embedding process, an improved digital hiding algorithm in the DCT domain is applied, wherein blocks that are more suitable for embedding are assigned larger embedding coefficients and greater modification magnitudes, thereby balancing the watermark's robustness and imperceptibility. Experiments are then conducted to validate the algorithm's reversibility and to analyze the effects of different embedding strengths on both robustness and imperceptibility.

Keywords: Digital Watermark, QR Code,Chaotic System,HVS, Discrete Cosine Transform

插图索引

符号对照表

符号	符号名称
$(d_A, P_A), (d_B, P_B)$	长期密钥对
$(k_A, K_A), (k_B, K_B)$	临时密钥对
$Cert_A, Cert_B$	数字证书
ID_A, ID_B	身份标识
r_A, r_B	随机数
R_A, R_B	随机数基点乘值
SN_A, SN_B	数据包序号
G	椭圆曲线的基点
n	椭圆曲线基点的阶
$DPTKM_A, DPTKM_B$	传输密钥材料
PRK	伪随机密钥
p	私有字符串
s	随机盐值
c	上下文信息
$KDF(\cdot)$	密钥派生函数
$DPTK_A, DPTK_B$	会话密钥
M, M', M_1, M_2	消息
$E(\cdot)$	加密函数
$Sig(\cdot)$	签名函数
$MAC(\cdot)$	消息认证码函数

表格索引

缩略语对照表

缩略语	英文全称	中文对照
DCT	Trusted Execution Environment	可信执行环境
HVS	Software Guard Extensions	软件保护扩展
DWT	Functional Encryption	函数加密
DFT	Key Management Enclave	密钥管理飞地
PDF417	Portable Data File 417	便携数据文件
QRCode	Quick Response Code	快速答复码
KGC	Key Generation Center	密钥生成中心
KDF	Key Derivation Function	密钥派生函数
MAC	Message Authentication Code	消息认证码
PBKDF	Password-Based Key Derivation Function	基于口令的密钥派生函数
PRF	Pseudo-Random Function	伪随机函数
SCKDF	Stream Cipher-based Key Derivation Function	基于流密码的密钥派生函数
MFKDF	Multi-Factor Key Derivation Function	多因子密钥派生函数
RSK	Root Seal Key	根密封密钥
IAS	Intel Attestation Service	英特尔认证服务
LA	Local Attestation	本地认证
RA	Remote Attestation	远程认证
RSK	Root Seal Key	根密封密钥
IAS	Intel Attestation Service	英特尔认证服务
HMAC	Hash-based Message Authentication Code	基于哈希函数的消息认证码
ECC	Elliptic Curve Cryptography	椭圆曲线密码学
CMAC	Cipher-based Message Authentication Code	基于分组密码的消息认证码
EPC	Enclave Page Cache	飞地页面缓存

目录

第一章 绪论

1.1 研究背景及意义

随着信息技术的迅猛发展，数字图像已成为网络信息传播的核心载体之一，有越来越多的图片需要存证，常见的处理方法是将图片哈希值和元数据传输给第三方存证系统存证，这样就能利用避免将信息保存在第三方的面临的数据泄露风险和效率低下的问题。

本文在存证哈希值和元数据的基础上，还可以为为存证体系引入新的安全机制，即存证系统根据用户提供的图像哈希值，元数据和用户自定义密钥生成和原始信息对应的唯一数字水印和该数字水印对应的嵌入算法，用户在收到存证机构的数字水印和与之对应的嵌入算法后，按照嵌入算法将数字水印插入到原始图片当中并保存。

为了解决让数字水印包含图片的版权信息并且防止版权信息被窃取的问题，本文设计了结合混沌系统和二维码的数字水印的生成和加密方案，该加密方案在保护用户版权的同时，防止泄露用户的版权信息。该方案将 QR 码和混沌系统结合，将用户上传的图片哈希值和元数据信息利用 SHA256 算法生成唯一的散列值 S。并将散列值 S 生成为 QR 码。然后将 QR 码利用混沌系统和 DNA 编码扩散技术继进行置乱-扩散操作。这样水印图片就包含了图片的版权信息，同时只有通过加密密钥才能解密该版信息。

为了平衡数字水印鲁棒性和不可见性，以及防止数字水印被非法提取，本文设计了结合人类视觉系统（HVS）和混沌系统的 DCT 域二值图像嵌入算法。该算法在兼顾水印鲁棒性和不可见性的同时，让水印在没有用户密钥的情况下无法被提取。具体做法是：将原始图片进行分块，依据人类视觉系统对图片边缘复杂度、纹理复杂度、亮度的不同敏感性，给每块图像打分并排序，分数越高的图像块越适合做水印的嵌入，越适合嵌入信息的图像块嵌入系数越大，修改幅度越大，以此来平衡数字水印的鲁棒性和不可见性。随后将二值数字水印序列化成 0，1 比特序列，并结合用户提供的密钥和混沌系统随机确认每个比特嵌入哪个图片块，这样做防止其他人在没有用户密钥的情况下想提取出数字水印并破解，保障了数字水印的安全性。然后基于改进的 DCT 域的加法嵌入方式进行嵌入操作。最后进行实验，验证算法可逆性，分析在不同嵌入强度下对鲁棒性和不可感知性的影响。

1.2 国内外研究现状

1.2.1 混沌系统加密的发展现状

混沌密码学的发展历程可追溯至20世纪非线性科学的突破性发现。1963年，Lorenz[1]

在气象动力学研究中首次揭示确定性系统的初值敏感特性，其构建的三维常微分方程组为混沌理论奠定了数学模型基础，这项奠基性工作被公认为混沌科学诞生的标志。

21 世纪以来，简单的低维度混沌系统被大量的运用到混沌加密中来，低维混沌系统凭借结构简洁性与实现高效性，在图像加密领域持续发挥重要作用[2]。然而，传统 Logistic 映射等一维系统存在的密钥空间受限、Lyapunov 指数偏低等缺陷，容易被预测[3]。所以在低纬度的混沌系统的研究方面，学者们提出了一些难以被预测但是结构相对简单的低纬度混沌系统。Logistic 系统作为经典的一维混沌系统被很多学者改造，具体有：Hua 团队[4]将 Logistic 混沌系统和 sine 映射结合起来，组成了一种二维混沌的系统，但是这种混沌的系统计算成本比一维系统要大很多。Zhou 团队[5]结合了 logistic 和 sine 两个一维混沌系统，提出了新的一维混沌映射结构，并将该结构应用到了图像加密领域。

随后研究者通过构建多维复合混沌系统显著提升了加密系统的密钥空间维度。混沌系统研究呈现从低维向高维的技术演进路径。此类的高维度的混沌系统比较经典的有 Chen 系统，Lorenz[6]系统等。Kuate 团队[7]就以 Loren 混沌系统为基础，提出了一个没有平衡记忆的新的混沌系统。包涵团队[8]提出了一种二维混沌映射，这种映射代数结构简单，通过改变初始条件可以控制映射的振幅，比较适合一些基于混沌映射的工程应用。

由于混沌系统具有一些密码学的特性所以混沌系统也是图像加密算法重要的一个环节[9]。混沌图像加密机制的设计质量直接影响算法的综合性能，其核心在于平衡安全强度与运算效率这对矛盾指标。当前主流方法普遍采用"位置置乱-数值扩散"的双阶段架构，研究者们通过优化各阶段的操作粒度和动态特性来提升整体加密效能。根据操作单元的不同，现有技术主要分为两大实现范式，像素级加密和比特级别加密：像素级加密以单个像素为基本处理单元，通过混沌序列驱动的坐标变换（如循环移位、矩阵转置）实现快速置乱。比特级加密则深入至像素的二进制位层面，采用位平面分解、DNA 编码[10]等微观操作实现精细扰动。

像素级加密通过改变像素的位置或值实现图像置乱，具有计算效率高、实时性强的特点。研究者普遍采用一维或低维混沌系统（如 Logistic 映射）生成置乱序列。Huang 等[11]提出的基于 Logistic 映射的像素置乱算法，通过混沌序列重排像素位置，破坏图像空间相关性。针对低维混沌系统密钥空间小的问题，Wang 等[12]设计了二维 sine-logistic-tent-coupling 映射（2D-SLTC），通过双向锯齿遍历增强像素置乱效果，显著降低相邻像素相关性。Fridrich 架构[13]是像素级加密的经典范式，Li 等[14]采用“置换-扩散”两阶段结构，利用改进的 Henon 映射实现像素位置和值的双重置乱，并通过扩散操作提升抗统计攻击能力。以上是混沌系统在像素加密的一些研究，像素级加密算法复杂度低，适用于实时传输场景（如遥感图像），但其仅改变像素位置或值，难以抵御针对统计分布的已知明文攻击。

比特级图像加密通过操作像素的二进制位实现信息混淆，其核心在于微观层面的比

特扰动与宏观统计特性的协同优化，安全性更高，但是计算成本较大。在国内研究创新中 Liu 等[15]提出的 DNA 编码与双混沌系统结合方案，将像素转换为 DNA 序列进行异或运算，实现比特级扩散。Qian 团队[16]引入忆阻混沌系统与双向比特循环移位，动态调整位平面排列，并通过 DNA 编码规则增强随机性。Hua 等[17]提出多比特置换与扩散（MBPD）框架，以 3 到 7 位为处理单元，结合 4D 超混沌系统生成掩码，较传统 1 位操作提升抗差分攻击能力。J Zhang[18]使用全局比特置换（GBCS）通过 SHA-256 哈希生成密钥，结合 4D 超混沌序列实现位平面全局混淆。

1.2.2 数字水印技术的发展现状

数字水印技术主要是针对空域或变换域来将图片中某一些比特位进行修改。由于变换域水印鲁棒性更强，本文选择变换域水印来实现功能，所以主要介绍变换域水印的技术发展。本文在实现数字水印嵌入算法中涉及水印加密和人类视觉系统，所以还会介绍一些水印加密算法以在 HVS 融合数字水印的算法。

空域上最经典的方法就是最低有效位（Least Significant Bit，LSB）[19]。该水印方法的有点就是原理简单，可以隐藏信息的容量大。但是缺点也显而易见，由于是在空域修改的信息，所以鲁棒性较差，嵌入的信息容易丢失，而且也容易被 StegExpose[20]检测。虽有也有一些空域的隐写技术被提出比如和 HILL[21]技术，WOW[22]技术，S-UNIWARD[23]技术等，这些技术会在嵌入的时候考虑视觉系统对画面的敏感性，自适应嵌入以降低视觉失真的风险，但是这些方法的鲁棒性比变换域嵌入低。

变换域技术一般是将图像变换到频率域，然后改变图像的某一些频域系数来完成对水印的嵌入，嵌入完成后进行逆变换就能得到载密图像，变换域水印技术的嵌入容量和空域比起来小一些，但是不可见性以及鲁棒性具有优势。现在比较主流的频域数字水印嵌入算法有离散小波变换（Discrete Wavelet Transform，DWT）[24]以及离散余弦变换（Discrete Cosine Transform，DCT）[25]技术，离散傅里叶变换（DiscreteFourier Transform，DFT[26]。比如 Xu 团队[27]将图像 8×8 分块后做 DCT 变换，在嵌入图像是用密钥随机的选取要改变的中频系数，这样提高了抵抗分析检测的能力，但是可嵌入的信息量较少。Kundur 团队[28]基于离散小波变换提出了按照小波分解变换层次的自适应数字水印算法，在水印嵌入的时候依照人眼视觉系统（HVS）动态设置水印强度来优化水印的不可见性。Ariatmanto 团队[29]提出了一种基于 DCT 的自适应缩放因子的水印嵌入算法，算法选取最大方差的图像块作为嵌入区域，并将 DCT 系数矩阵中的中频系数按照最优比例的嵌入因子完成嵌入操作，该算法在保障不可见性同时很好抵抗滤波攻击，噪声添加。

水印加密算法：水印加密算法融合和将数字水印和图像加密技术结合，旨在保护水印信息的版权性。Liu 团队[30]将 logistic 混沌系统的系数用传统密码学中的 RSA 加密保护，再利用混沌系统对水印进行加密，保护水印的同时增加了鲁棒性。马婷团队[31]设计

了一种基于 NSCT-DWT-SVD 的彩色图像水印加密技术，首相将数字水印编码成二维码，对二维码采用 logistic 混沌系统进行加密，最后嵌入到载体图像的奇异值，实验测试加密算法对集合攻击有良好的抵御。周希团队[32]结合了 Tent 映射和 Sinusoidal 映射，并将构造的新的映射用来把图像的位置信息置乱，最后选择载体图像的中频系数总嵌入。

HVS 和数字水印的结合：刘伟宏团队[33]提出了一种自适应嵌入强度的数字水印算法，算法通过图像分块的纹理和亮度特征来自适应调整嵌入强度。沈磊等[34]提出了一种基于 HVS 将图像分块分类的的嵌入算法，具体是将图像分为平滑、纹理和边缘 3 个区域，每个不同的区域嵌入强度不同。

1.3 主要研究内容

主要研究内容如下，研究内容之间的关系如图 1.1 所示：

（1）为了解决让数字水印包含图片的版权信息并且防止版权信息被窃取的问题，本文设计了结合混沌系统和二维码的数字水印的生成和加密方案，该加密方案在保护用户版权的同时，防止泄露用户的版权信息。该方案将 QR 码和混沌系统结合，将用户上传的图片哈希值和元数据信息利用 SHA256 算法生成唯一的散列值 S。并将散列值 S 生成为 QR 码。然后将 QR 码利用混沌系统和 DNA 编码扩散技术继进行置乱-扩散操作。这样水印图片就包含了图片的版权信息，同时只有通过加密密钥才能解密该版信息。

（2）为了平衡数字水印鲁棒性和不可见性，以及防止数字水印被非法提取，本文设计了结合人类视觉系统（HVS）和混沌系统的 DCT 域二值图像嵌入算法。该算法在兼顾水印鲁棒性和不可见性的同时，让水印在没有用户密钥的情况下无法被提取。具体做法是：将原始图片进行分块，依据人类视觉系统对图片边缘复杂度、纹理复杂度、亮度的不同敏感性，给每块图像打分并排序，分数越高的图像块越适合做水印的嵌入，越适合嵌入信息的图像块嵌入系数越大，修改幅度越大，以此来平衡数字水印的鲁棒性和不可见性。随后将二值数字水印序列化成 0，1 比特序列，并结合用户提供的密钥和混沌系统随机确认每个比特嵌入哪个图片块，这样做防止其他人在没有用户密钥的情况下想提取出数字水印并破解，保障了数字水印的安全性。然后基于改进的 DCT 域的加法嵌入方式进行嵌入操作。最后进行实验，验证算法可逆性，分析在不同嵌入强度下对鲁棒性和不可感知性的影响。

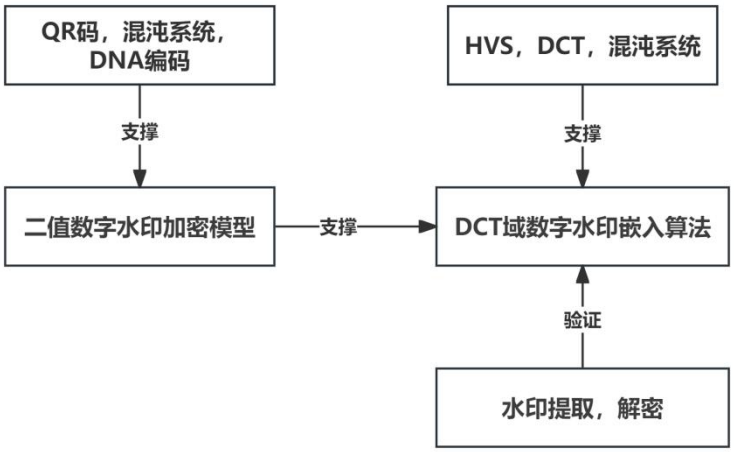


图 1.1 研究内容之间的关系

1.4 论文组织结构

论文的章节安排如下：

第一章 绪论部分，本章节主要是介绍了论文的研究背景和研究意义，阐述当前图片存证中可以优化的部分，并提出了自己的解决方案，系统梳理混沌系统、数字水印技术领域的研究进展，明确研究目标与技术路线。

第二章 相关理论基础：介绍了二维码的分类和构成，重点介绍了混沌系统理论和数字水印技术。对第三、第四章的算法做好理论准备。

第三章 基于混沌系统的数字水印加密加密算法。介绍了用户上传的图像哈希、元数据和密钥，如何转换成二维码数字水印，以及混沌系统结合洗牌算法和 DNA 编码技术将二值数字水印如何置乱-扩散。最后做实验验证加密算法的鲁棒性以及置乱-扩散效果是否良好。为第四章的水印嵌入做好准备。

第四章 基于 HVS 的 DCT 域数字水印加密嵌入算法：讲解在第三章生成数字水印后如何将水印加密嵌入载体图像，具体介绍了：如何利用 HVS 将载体图像分块打分算法，该算法可以量化图像块是否适合嵌入水印。如何兼顾鲁棒性和不可见性的在 DCT 域嵌入水印，以及如何将水印加密嵌入载体图像。最后做实验验证不同嵌入强度对鲁棒性和不可见性的印象。

第五章 总结和展望。对全文已经完成的工作进行了总结，并提出了还可以改进的工作。

第二章　背景知识

2.1 二维码技术

二维码技术本质上是基于二维空间几何图案的信息编码体系，其通过明暗色块的矩阵式分布实现数据表征。在编码机制层面，设计者利用明暗模块的光学对比特性（通常采用深色/浅色组合）对应二进制数据流中的逻辑值，这种映射关系使得光电传感装置可通过识别模块的空间拓扑结构解析出原始信息。现代解码系统通常集成模式识别算法与纠错编码技术，能够自动处理模块几何变形、局部遮挡等复杂情况。值得关注的是，不同编码规范（如 QR Code、PDF417 等）通过差异化模块布局策略实现分级的容错机制，例如 QR 码的 Reed-Solomon 编码可支持高达 30%的数据恢复能力。当前该技术已深度渗透至商业生态的各个环节：在消费领域支撑移动支付（如支付宝/微信扫码）、在物流管理实现全链条追溯（GS1 标准应用）、在公共安全领域用于证件防伪（公安部电子标识系统），并在智慧城市建设中承担空间位置服务载体功能（腾讯地图街景编码）。特别在新冠疫情防控期间，健康码系统的全国性部署更凸显了二维码技术在数据实时交互与可信认证方面的独特价值。

2.1.1 二维码的分类

（1）堆叠式二维码

堆叠式二维码属于复合层叠结构的编码体系，通过纵向压缩一维码并进行多层堆叠实现数据扩容。该技术采用模块化组合架构：将传统线性条码的纵向尺寸缩减后，通过垂直方向的多层叠加形成矩阵式数据载体。其核心特征在于既保留了一维码的可识别性，又实现了二维空间的信息扩展。

技术实现层面，每个堆叠层实质上构成独立的一维编码单元，这使得常规条码读取设备仍能实现基础识别功能。但因其特有的垂直排列结构，系统需配备多层解码算法来识别堆叠层数并实施复合解析。这种解码机制包含行序判定、层间数据重组等特殊处理流程，与普通一维条码处理技术存在显著差异。典型应用实例包括 PDF417、Code 16K 及 Code 49 等国际主流复合码制。

该编码体系具备显著的技术优势：信息密度方面，最大可承载 1800 个英文字符或 2700 位数字代码，纠错能力采用 RS 冗余校验技术，支持用户自定义容错级别（最高可达数据损毁 30%仍可复原），符号结构遵循模块化设计原则，符合 GB/T18284-2000 国标规范要求[35]。

以 PDF417 码为例，其编码结构具有典型代表性。每个字符单元由 4 个条纹和 4 个

间隙组合而成，共包含 17 个标准单元（即"Portable Data File 417"的命名由来）。这种特殊构造使其兼具高密度存储和强抗损特性，被广泛应用于证件防伪、物流追踪等领域[36]。国家标准不仅明确定义了其物理尺寸、符号构造等基础参数，还对印刷质量、解码规则等实施严格技术规范。如图 2.1 所示

图 2.1　pdf417 示例

（2）矩阵式二维条码

矩阵式二维条码（2D Matrix Code），又称棋盘式二维码，是一种通过几何图形空间分布实现信息编码的符号系统。其技术核心是将二进制数据映射为黑白模块的矩阵排列——黑色模块对应"1"，白色模块对应"0"，利用模块的位置、比例和组合关系构成数据载体。作为组合编码与图像处理技术融合的产物，此类码制具备高密度存储、容错性强和快速识读等特性，典型代表包括 QR Code[37]、Data Matrix、Code One 及 Maxi Code 等。从结构学角度分析，矩阵式二维码以中心定位点为基准，通过辐射状多边形单元构建功能图形（如定位标志、校正模式）与编码区域的分层结构，前者确保扫描设备的空间定位与畸变校正，后者则通过模块化排列存储数据内容。其中 QR 码作为最广泛应用的标准，由日本电装公司（Denso Wave）于 1994 年研发并开放专利，其技术规范包含 40 个版本规格，最高可存储 7089 个数字或 4296 个字符信息，通过纠错算法实现最高 30%的数据恢复能力。国际标准化组织（ISO）在 2000 年将其纳入 ISO/IEC 18004 标准[38]，标志着该技术进入规模化工业应用阶段，目前已在物流追踪、移动支付、智能制造等领域形成完整的生态系统。

QRCode 码（Quick Response Code）如图 2.2 所示：

图 2.2 QR 码示例

2.1.2 QR 码的构成

本文采用了矩形二维码中的 QR 码，所以接下来着重介绍 QR 码的相关知识。如图 2.3 所示 QR 码由两个区域构成[39]，分别是编码区域和功能区域，下面介绍这两个区域。

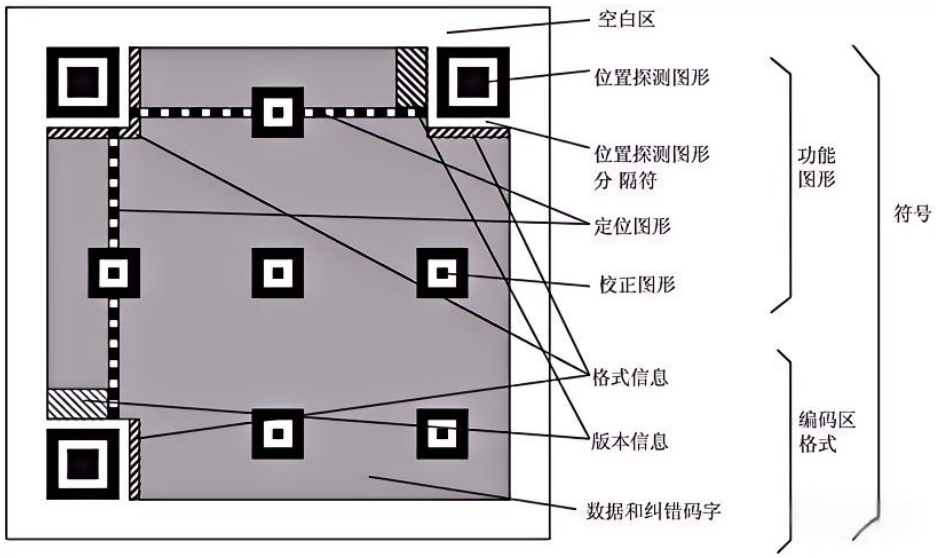


图 2.3 QR 码构成图

(1)功能图形

功能图形作为 QR 码的结构基准由寻像图形、定位图形、校正图形和分隔符号构成。其中，由三层同心方框组成的寻像图形通过特殊的黑白比例（1:1:3:1:1）实现快速定位，等间隔黑白条纹的定位图形建立坐标系基准，按固定间距排列的校正图形矩阵支持不同版本 QR 码的形变校正，而环绕寻像图形的白色分隔符号则确保功能区域与数据区域的清晰隔离。这些图形元素在不同版本和编码数据中都保持固定的几何形态，为 QR 码的快速识别和精确解码提供空间基准。与之对应的编码区域采用可变结构设计，其模块化排列的数据码字携带核心信息，纠错码字通过里德-所罗门算法实现数据容错修复，格式信息存储纠错等级与掩模模式参数，版本信息则记录 QR 码规格标识。该区域的二进制数值将根据输入内容、版本尺寸、纠错等级（L/M/Q/H）等参数动态生成，形成既包

含用户数据又具备容错能力的完整编码体系。

(2)编码区

QR 码的编码区域采用模块化动态编码机制，由数据码字、纠错码字、格式信息与版本信息四类核心组件构成。其中，数据码字通过模式指示符（数字/字母/字节等编码模式）将输入信息转化为 8 位二进制序列，并按字节块进行分组存储；纠错码字基于里德-所罗门算法生成冗余校验数据，可根据预设的纠错等级（L:7%/M:15%/Q:25%/H:30%）实现受损模块的数学重建；格式信息通过 15 位编码记录纠错等级与掩模模式参数，其数据通过双通道嵌入在定位图形附近，确保任意方向读取的鲁棒性；版本信息则在版本 7 以上 QR 码中显式存在，采用 18 位二进制编码记录版本号，并通过 BCH 纠错码生成校验位，沿寻像图形外围形成特定几何排列。整个编码区域遵循 ISO/IEC 18004 标准，其模块的二进制状态由数据内容、版本规格、纠错参数及掩模运算结果动态决定，最终构建出兼具信息承载能力与容错冗余度的二维矩阵结构。

2.2 混沌系统

混沌作为普遍存在于非线性系统中的复杂动力学行为，其典型特征表现为确定性系统内蕴的类随机特性。1963 年，麻省理工学院气象学教授 Edward Lorenz 通过大气动力学研究首次建立了混沌系统的数学模型，并阐释了具有里程碑意义的"蝴蝶效应"（Butterfly Effect）。随着非线性科学的纵深发展，混沌系统在信息安全（如量子密钥分发）、先进制造（如半导体激光器优化）、空天科技（如航天器姿态控制）等前沿领域展现出独特的应用价值。这种兼具确定性机制与不可预测性的特殊性质，使得混沌理论得以与量子力学、相对论共同构成现代物理学的三大支柱理论[40]。

2.2.1 混沌理论

1975 年，应用数学家李天岩（T. Y. Li）与其导师 James A. Yorke 在《美国数学月刊》上开创性地构建了混沌的数学分析框架，提出具有奠基性意义的"周期三蕴含混沌"定理（Period Three Implies Chaos），并建立了被学界广泛采纳的 Li-Yorke 混沌定义[41]。该定义因其严格的数学表述而成为混沌研究领域最具普适性的判定准则之一，其形式化描述如下：

设 *f(x)*是在为闭区间 L 上的连续自映射函数，若满足下面的条件，就可以说函数 *f(x)* 说在不可数子集 *S* 上是混沌函数。

(1) 系统周期点的周期构成无界集合

(2) 存在不可数子集 S⊂L 且 S 不包含周期点，满足：

对任意 $x, y \in S$，满足式(2.1)：

$$\lim_{n\to\infty}\inf\left|f^n(x)-f^n(y)\right|=0 \tag{2.1}$$

对任意 x, y∈S，当 x≠y 时，满足式(2.2)：

$$\lim_{n\to\infty}\sup f_n(x)-f_n(y)|>0 \tag{2.2}$$

对任意x∈S 和在函数任意周期点 y，满足式(2.3)

$$\lim_{n\to\infty}\sup f_n(x)-f_n(y)|>0 \tag{2.3}$$

基于 Li-Yorke 定义的数学框架，混沌系统展现出独特的非线性动力学特征，其主要特性可归纳如下[42]：

相空间约束性：尽管表现出非周期运动模式，系统的动力学行为始终被限制在相空间的特定拓扑结构（即混沌吸引子）内。

遍历性：混沌域内所有的状态值该系统在一段时间内都会经过。

内生随机性：系统表现出的随机行为源自其非线性耦合的内在特性，与外部随机扰动存在本质区别。

初值条件敏感依赖性：系统轨迹具有指数发散特性，即使初始条件相差很小但是最终的系统轨迹也完全不同。

预测视界有限性：系统状态预测误差随时间呈指数增长，导致有效预测时间窗口极短，没有长期的预测性。

普遍实用性：系统特征是系统内在的规律体现，不会参数和动力方程改变而变化。

2.2.2 混沌效果的判定体系

混沌的判定不是一种绝对的方法，而是一种相对的方法。不同的混沌系统可能需要不同的判定方法，而且有时候混沌行为可能只是临时的，系统在不同条件下可能表现出不同的行为。因此，综合多种方法和指标来判断混沌是通常的做法。以下是一些常见的混沌判定方法：

(1) Lyapunov 指数（Lyapunov exponent）[43]：Lyapunov 指数表用来量化系统是混沌运动的重要指标：一个稳定系统的最大 Lyapunov 指数为负数，而混沌系统中至少存在一个正的 Lyapunov 指数。而且 Lyapunov 指数还可以判断一个系统是否是更为复杂的超混沌系统，当系统存在两个或两个以上的正 LE 值，那么这个系统为超混沌系统。Lyapunov 指数表达式如式(2.4)所示：

$$LE = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln|f'(x_i)| \tag{2.4}$$

(2) 分岔图：分岔图是一种可视化方法，通过观察系统参数变化时轨迹的分支模式来判断混沌。当参数变化引起轨迹的分支和分叉现象时，系统可能呈现混沌行为。

(3) 庞加莱截面：庞加莱截面是在相空间中选择一个特定的平面，观察轨迹与该平面的交点。如果交点的分布呈现复杂的非周期性特征，那么系统可能是混沌的。

(4) 分维数计算：通过计算系统的分维数，可以评估系统的复杂性。高分维数通常与混沌系统相关联。

2.2.3 经典的混沌系统

(1)经典一维Logistic 系统

经典一维Logistic 系统其动力学方程如式(2.5)所示：

$$x_{n+1} = \mu x_n(1-x_n) \tag{2.5}$$

其中 μ 为系统的控制参数，其取值范围为 $\mu \in (0, 4)$ ，此时 Logistic 映射为混沌状态，会生成混沌序列x_n，且$x_n \in (0, 1)$ 。

(2)Tent 映射

Tent 也是一维混沌映射，该映射是一种分段线性映射，Tent 映射的数学结构简单、函数均匀的分布，Tent 映射被广泛用于混沌加密系统中。Tent 映射描述如式(2.6)所示：

$$x_{n+1} = \begin{cases} \mu x_n & 0 < x_n < 0.5 \\ \mu(1-x_n) & 0.5 \le x_n \le 1 \end{cases} \tag{2.6}$$

其中参数 $\mu \in (0,2]$，当 $\mu \in (0,1)$时系统会逐渐收敛到 0，不具有混沌特性。 但当 $\mu \in (1,2)$时，Tent 映射会出现周期性行为和分岔，μ=2 时，系统表现为完全混沌状态，具有高度的初值敏感性。

（3）Henon 映射

Henon 映射是一个的二维的混沌系统，其映射方程如式(2.7)所示：

$$\begin{cases} x_{n+1} = 1 + y_n - ax_n^2 \\ y_{n+1} = bx_n \end{cases} \tag{2.7}$$

系数 a，b 为系统参数，当 b=0.3，$a\in(1.06,1.22)\cup(1.27,1.29)\cup(1.31,1.42)$时，系统处于混沌状态，相比于之前介绍的一维映射相比，Henon 映射有 2 个系统参数，混沌空间也更大。

（4）Chen 系统

陈氏混沌系统是由美国休斯顿大学的陈关荣教授在 1999 年首次提出的。陈关荣教授是混沌理论和非线性动力学领域的重要学者，他在探索与著名的 Lorenz 系统不同的混沌吸引子时，发现了这一系统[44]。

为了寻找与 Lorenz 系统不同但同样能产生复杂混沌行为的系统，陈关荣教授提出了陈氏系统。该系统不仅具有与 Lorenz 系统类似的初值敏感性和非周期性，而且在拓扑结构上与 Lorenz 系统不等价，为混沌吸引子研究提供了新的范例。Chen 系统的数学表示式如式(2.8)所示。

$$\begin{cases} \dfrac{dx}{dt} = a(y-x) \\ \dfrac{dy}{dt} = (c-a)x - xz + cy \\ \dfrac{dz}{dt} = xy - bz \end{cases} \tag{2.8}$$

其中，x 、y 和 z 是系统的三个状态变量，a、b、c 是系统参数。Chen 系在 $a=35$，$b=3$，$c=28$ 表现出的混沌状态。陈氏混沌系统在需要高度随机性和不可预测性的场景中具有明显优势。

2.2.4 结合混沌系统的图像加密流程

数字图像加密技术旨在将图像转换为类似噪声的形式，以便在传输过程中保护其内容的安全性，并确保在接收端能够恢复原始图像。其基本原理是将数字图像视为一个与其尺寸相同的像素矩阵。由于图像内容随着像素的变化而改变，因此，图像加密的关键在于如何有效地改变图像的像素位置和像素值。

Shannon 很早就指出了密码系统的两个原则：混淆和置乱[45]。混沌数字图像加密技术结合了混沌密码学和数字图像加密技术，利用混沌序列发生器生成的混沌序列，对明文图像在像素平面或位平面进行置乱和扩散操作，从而改变明文图像的像素状态，最终实现图像的加密。以下是混沌图像加密的详细流程：

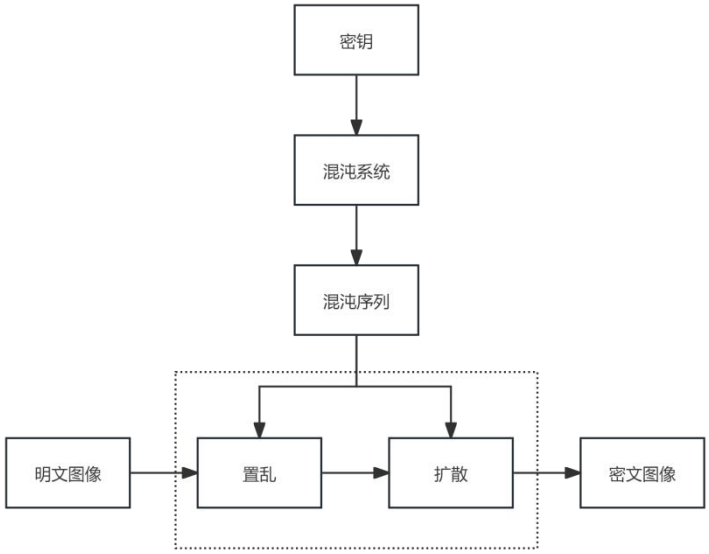


图 2.4 混沌加密流程

（1）预处理：这一步是对原始图像进行处理，比如可以将图像从 RGB 图像转换成灰度图形，或者将原始图像转换成二值图像。

（2）获取密钥：通过随机算法获取加密的密钥。

（3）密钥生成混沌序列：根据实际的需求，在考虑安全性和加密速度的情况下，选择合适的混沌系统，如 Logistic 映射或 Henon 映射等。利用已经获取的加密密钥，通过迭代生成所需的混沌序列，用于加密过程。

（4）对原始图形进行扩散和混淆：将原始图像按照一定的规则划分成不同的区域，对不同的区域逐个进行处理，或按逐像素方式加密。对每个像素或像素块执行相应操作，混沌序列用于控制像素位置的重排和亮度或颜色分量的替换。

（5）生成载密图像：完成扩散与混淆操作后，得到载密图像。将其保存为新文件，供传输或存储。

（6）解密载密图像：使用相同的混沌系统、密钥和参数，按相反顺序执行解密操作：恢复像素位置、还原亮度或颜色值，并转换回原始颜色空间（若在加密前已做转换）。解密后图像应与原始图像高度相似。

（7）对加密算法进行测试评估：对加密算法进行各种攻击测试（如差分攻击、已知明文攻击等），评估其安全性。根据测试结果调整混沌系统参数，改进扩散与混淆方法，并优化加密解密性能。

2.2.5 混沌加密的技术

（1）基于混沌的置乱方法

Arnold 置乱法：

适用于图像边长为 M 的正方形图像，首先将图像装换成 M×M 的矩阵，最后按照式(2.9)所表示的映射关系将原始图像中的像素关系进行置乱操作，置乱后的图像的像素处于混沌状态。

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (modM) \tag{2.9}$$

其中，原始图像的横坐标和纵坐标分别用 x 和 y 来表示，加密图像的横坐标和纵坐标分别用x'和y'来表示，矩阵式 M×M 大小。

（2）基于混沌的扩散方法

直接运算法：将明文图像作为像素矩阵进行处理，首先生成与像素矩阵同维度的混沌矩阵。对混沌矩阵元素依次执行数值放大、整型化及模运算处理，形成随机混沌矩阵。随后通过像素矩阵与随机混沌矩阵之间的模加运算或按位异或运算实现加密，最终输出密文图像。整个加密过程的核心在于通过数学变换改变原始像素值，其中模运算参数需与图像位深度保持匹配以确保数值有效性。

DNA 编码运算法：基于生物 DNA 碱基互补特性建立数字编码系统，四种核苷酸分别对应二进制编码（A=00、T=11、C=01、G=10），构成八种可选的互补编码规则。如表 1-1 所示，每个编码规则不仅定义碱基与二进制的映射关系，同时满足"00-11"、"01-10"两对互补组合的对应转换。在加密过程中，图像像素值（如 158 对应二进制 10011110）与混沌序列数值分别通过选定编码规则转换为 DNA 链（以规则 1 为例生成"GCTG"），随后基于 DNA 运算表（如表 2.2 的加法规则）对两组 DNA 序列执行算术或逻辑运算，最终通过逆编码还原为加密像素值。该算法通过双重编码转换与生物运算机制实现像素值扩散，其加密强度取决于编码规则与运算规则的组合选择。

表 2.1 DNA 编码规则

	规则 1	**规则 2**	**规则 3**	**规则 4**	**规则 5**	**规则 6**	**规则 7**	**规则 8**
00	A	A	T	T	C	C	G	G
11	T	T	A	A	G	G	C	C
01	C	G	C	G	T	A	T	A
10	G	C	G	C	A	T	A	T

表 2.2DNA 加法运算

加法	A	G	C	T
A	A	T	C	G
T	T	G	A	C
C	C	A	G	T
G	G	C	T	A

表 2.3DNA 减法运算

减法	A	G	C	T
A	A	G	T	C
G	G	A	C	T
C	C	T	A	G
T	T	C	G	A

表 2.4DNA 异或运算

异或	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

2.3 数字水印

数字水印技术实施流程包含三个核心环节：预处理阶段对水印信息实施加密相关处理；载体图像通过频域变换处理构建水印嵌入空间；选择特定嵌入策略完成信息融合。常见嵌入方法分为空域与频域两类，例如在空域中通过调整像素值产生细微差异实现信息隐藏。水印提取需逆向执行嵌入算法，其鲁棒性体现在载体图像经历常规信号处理（如压缩、滤波）后仍能有效提取水印信息。系统性能通过不可见性（嵌入前后图像视觉一致性）与鲁棒性（抗攻击能力）双重指标进行量化评估，二者共同构成水印方案有效性的核心判别标准。

2.3.1 数字水印的基础模型

数字水印技术指通过特定算法在载体图像中隐蔽嵌入标识信息（如二进制序列、数字签名或生物特征数据）的数字版权保护方法。其系统架构包含嵌入与提取两大核心环节：嵌入阶段将加密处理后的水印数据融合至载体图像的频域或空域特征中，要求保持载体视觉质量无明显劣化；提取阶段则通过逆向算法从可能遭受攻击的载体中恢复水印信息，实现版权溯源。系统有效性取决于不可感知性（视觉隐蔽度）与鲁棒性（抗压缩/滤波等攻击能力）的平衡优化。

图 2.5 就是广义的数字水印的添加和提取模型：

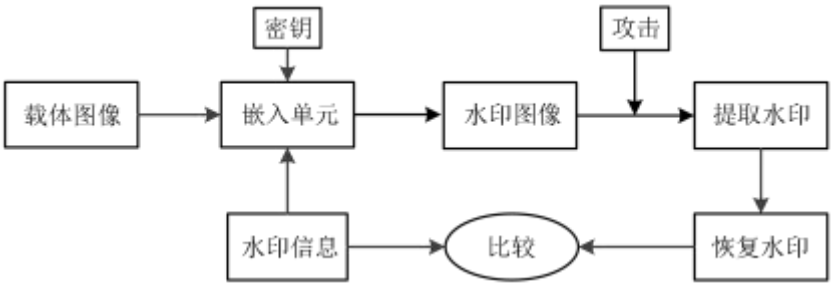


图 2.5 数字水印添加提取模型

嵌入单元：数字水印系统的嵌入单元由载体图像与水印信息构成基础输入要素，可选择性引入密钥机制以增强系统安全性。该单元的核心处理模块通过特定加密策略实现水印融合，其算法选择（如频域系数调制或空域像素调整）直接影响水印的隐蔽性与抗攻击能力。经算法处理后生成的含水印图像作为最终输出，其视觉质量需与原始载体保持高度一致性以确保不可感知性。

攻击单元：数字水印攻击指削弱水印可提取性的操作或干扰因素，主要分为非恶意干扰与恶意攻击两类。前者源于传输信道失真或常规信号处理（如 JPEG 压缩、噪声污染），后者则涉及针对性破坏手段。当前主流攻击可归纳为两大技术分支：1）信号处理攻击：通过频域/空域修改破坏水印结构，典型手段包括滤波、量化、重采样；2）几何攻击：通过空间变换扰乱水印同步机制，具体表现为旋转、缩放、裁剪等仿射变换[46]。鲁棒水印系统需建立双重防御机制：在算法层采用抗几何失真的同步标记设计，在数据层嵌入冗余纠错编码以抵御信号处理损伤。系统鲁棒性作为核心评价指标，直接决定水印在遭受复合攻击后的存活能力。

提取单元：该过程从水印图像或被攻击的图像中恢复隐藏的数据，通常是嵌入过程的逆操作。提取方法可以分为非盲、半盲和盲提取。非盲提取需要原始载体图像信息；半盲提取则依赖原始水印信息；而盲提取则不依赖任何先验知识。比较单元：比较单元是在最后的时候将提取出的水印与原始水印进行比较，提取后的水印和原始水印的相似

程度越高则该水印的鲁棒性越强。

2.3.2 数字水印的特性

数字水印有下面几个特性：

（1）不可感知性：

数字水印的不可感知性体现为含水印载体与原始载体在视觉特征上的高度一致性，要求人类视觉系统（HVS）无法察觉信息嵌入引发的失真。该特性在医学影像分析[47]、遥感[48]解译等敏感领域尤为重要，细微的视觉偏差可能导致诊断或判读错误。此外，视觉保真度的缺失会暴露水印存在，诱发针对性攻击（如定位擦除或覆盖篡改），致使版权保护机制失效。因此，不可感知性不仅是用户体验的基础要求，更是保障水印隐蔽性与功能有效性的核心约束条件。

（2）鲁棒性

数字水印的鲁棒性表征算法在遭受信号处理攻击（压缩、噪声干扰）与几何攻击（旋转、缩放、裁剪）时维持水印信息完整性的能力，其核心要求是通过冗余嵌入、频域能量扩展等技术手段，确保水印在攻击后仍具备可检测性与可恢复性。现行主流增强策略包括：1）频域扩频技术：将水印能量分散至宽频段以抵御局部信号损伤；2）几何同步机制：嵌入定位标记以校正空间变换引发的同步偏移；3）分层冗余编码：通过纠错码与多副本嵌入提升信息存活概率。高鲁棒性系统需实现攻击敏感性（快速识别篡改）与生存能力（维持水印完整）的动态平衡，从而有效阻止非授权方的水印擦除或篡改企图。

（3）嵌入容量

嵌入的容量指的是可以嵌入到载体图像中的水印数据量[49]。较大的嵌入容量能够存储更多的信息，但这也可能会影响水印的质量和鲁棒性。如果嵌入的容量过大，水印可能会变得容易受到攻击或者被察觉，从而影响不可见性和鲁棒性。

（4）安全性

安全性是指在没有授权的情况下，别人无法知道图片中嵌入了数字水印，或者即使知道嵌入了数字水印在没有授权（如密钥）的情况下可无法提取出数字水印。

鲁棒性与不可见性通常存在矛盾。为了提高鲁棒性，水印通常需要嵌入更多的信号，这可能会导致水印在图像中的可见性增加。因此，如何在保证鲁棒性的同时保持不可见性是一个挑战。鲁棒性与嵌入容量也有一定冲突。增加嵌入容量可以使得水印在受攻击后的恢复能力更强，但这也可能使得水印变得容易被检测或被修改，导致鲁棒性下降。不可见性与嵌入容量之间也有对立关系。增加嵌入容量可能会使得水印更加显眼，破坏不可见性。因此，需要在这三者之间找到一个平衡点。

2.3.3 水印嵌入技术的选择

目前数字水印技术主要采用的是空域数字水印技术和变换域数字水印技术。本节将介绍代表性的空域水印嵌入技术和变换域水印嵌入技术并且说明技术选择的依据。

1. 空域数字水印

空域隐写技术通过修改载体数据的非关键信息位实现信息隐蔽传输，其核心机制在于利用人类视觉系统（HVS）对细微亮度变化的低敏感特性。典型代表如最低有效位（LSB）算法，该技术将隐写数据嵌入像素值最低比特位，通过替换操作使载体修改量控制在±1 灰度级范围内，从而保证视觉不可感知性。LSB 方案具有两大显著优势：其一，隐写容量与载体像素数呈线性关系，可实现高数据吞吐；其二，算法复杂度低，适用于实时处理场景。但受限于底层嵌入机制，其对信号处理攻击（如 JPEG 压缩、重采样）表现出显著脆弱性——当载体经历有损处理时，LSB 层信息极易被量化过程破坏。研究指出，虽然空域方法在不可见性与嵌入容量方面表现优异，但需结合加密编码或信息分散策略才能提升抗攻击能力。

本文对数字水印嵌入技术的要求是具有一定的鲁棒性能抵御剪切，污损，压缩等，而空域技术对这些攻击的抵抗性太差，所以不选用空域数字水印技术。

2. 变换域数字水印

变换域隐写技术通过将秘密信息嵌入载体信号的频域系数实现隐蔽传输，其核心原理基于人类感知系统对频域能量分布的差异化敏感特性。相较于空域隐写，该技术通过频域能量调制策略（如中频带嵌入）在不可见性与鲁棒性之间取得更优平衡：1）低频分量修改易引发视觉失真，高频分量易受信号处理干扰，故选择中频区域作为信息载体；2）频域变换（DCT/DWT）的全局能量分布特性使嵌入信息具备抗局部裁剪、噪声干扰的能力。主流实现路径包括离散余弦变换（DCT）系数调制、小波域（DWT）子带能量调整以及压缩域隐写等复合技术。以图像载体为例，其技术流程可分为三步：首先对原始图像进行正交变换获取频域系数矩阵；随后根据隐写规则调整选定频段系数值；最后执行逆变换生成含密载体。该技术体系在抵抗 JPEG 压缩、滤波攻击等方面展现出显著优势，但需权衡计算复杂度与嵌入容量——频域变换的正交特性虽增强鲁棒性，却也限制了可修改系数数量。

（1）DCT 技术

数字水印技术中的 DCT（离散余弦变换）方法 通过将水印嵌入图像频域中频系数实现隐蔽性与鲁棒性的平衡。其核心原理是将图像分块，如 8×8 像素分块并进行 DCT 变换，将空域像素转换为频域能量分布——低频分量对应主体轮廓修改易失真、高频分量易受压缩破坏，因此选择中频区域作为水印嵌入位点。具体流程为：分块后对每个块执行 DCT 变换，通过量化步长控制修改选定中频系数，再逆变换重建含水印图像。水印提取时，可通过对比原始图像的非盲提取或直接解码频域系数的盲提取恢复信息。

DCT 技术的优势在于抗 JPEG 压缩能力强，与 JPEG 标准兼容、不可见性高，适合版权保护与内容认证；但面临几何攻击脆弱性，如旋转/裁剪，和容量限制（依赖分块数量）。典型应用包括 JPEG 图像版权标记、视频溯源追踪等。为优化性能，常动态调整强度结合自适应嵌入、混合域策略或抗几何模板进行改进，实现在压缩、滤波等常见攻击下的稳定水印存活。

（2）DWT 变换

数字水印技术中的 DWT（离散小波变换）方法 通过多尺度频域分解实现水印的隐蔽嵌入与高鲁棒性保护。其核心原理是将图像通过小波变换分解为多级子带（如 LL 低频子带、LH/HL 中频子带、HH 高频子带），选择中高频子带（如 LH 或 HL）嵌入水印——低频子带（LL）包含图像主体能量修改易导致失真，高频子带（HH）易被压缩破坏，而中频子带既保留细节特征又具备抗干扰能力。具体流程为：对原始图像进行多层小波分解，在中频子带系数中通过量化调制，如修改系数幅值或相位或系数替换嵌入水印，再通过逆小波变换重建含水印图像。水印提取时，需对含密图像执行相同的小波分解，从目标子带中解码水印信息盲提取依赖预定义规则，非盲提取可对比原始子带差异。

2.4 本章小结

本章主要介绍了后面要用到的一些理论知识。首先介绍了了二维码的分类，以及后面要用到的 QR 码的构成。紧接着介绍混沌系统的一些知识，列举了几个经典的混沌系统，介绍了混沌系统在图像加密方面的知识，让读者能提前了解混沌系统在图像加密的基础流程和技术。最后介绍了数字水印技术，了解数字水印基础模型，介绍数字水印的不可能三角，以及数字水印在空域和频域一些常用的技术手段和原理。这些知识为下面的基于混沌系统的水印加密方案和改进 DCT 域数字水印嵌入技术提供了理论基础。

第三章　基于混沌系统的数字水印加密算法

本章节将介绍存证系统获取元数据后，进行的主要操作包括：利用元数据生成原始图像的唯一标识，利用唯一标识码生成混沌系统初始值，将唯一标识编码成 QR 码，生成混沌序列，将 QR 码利用混沌序列进行混沌置乱和扩散操作得到置乱和扩散后的混沌图片。然后测试对混沌图片的反向提取操作，解密混沌图像得到 QR 码，最后进行安全性测试，测试混沌加密方案的有效性。

3.1 数字水印生成模型

加密系统的加密流程如图 3.1 所示。下面是对详细步骤的解释：

步骤一：通过明文图像提供的特征信息采用 SHA256 算法算出特征信息的哈希值。

步骤二：将生成的哈希值转换成 QR 码图像作为秘密信息。

步骤三：将哈希值作为加密密钥，对该散列值进行运算后得到混沌系统的初始值和参数值。

步骤四：混沌系统带入步骤三生成的初始值和参数值，生成用于置乱的混沌序列和用于 DNA 编码扩散的混沌序列。

步骤五：将洗牌算法和置乱混沌序列结合，对 QR 码图像进行置乱操作。

步骤六：将 DNA 循环编码扩散机制和扩散混沌序列结合，对步骤五得到的置乱图像进行扩散操作。

步骤七：经过步骤三，四，步骤五两个关键步骤后，即可完成对原始图像的加密处理并得到混沌图像。

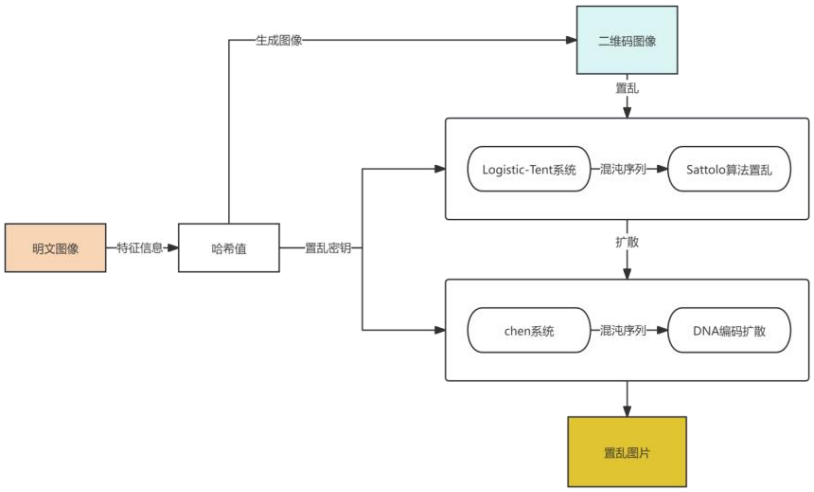


图 3.1 数字水印生成模型

3.2 混沌模型的选取

3.2.1 一维混沌体系的选取

在第二章的混沌系统中介绍了两个经典的混沌映射系统 Logistic 系统和 Tent 系统。Logistic 系统和 Tent 系统都是常见的混沌系统，但它们在动态特性上有所不同。Logistic 系统具有较简单的数学模型和较好的稳定性，但对于一些复杂的非线性现象，它的表现较为有限。Tent 系统则在产生更为复杂的混沌行为方面表现得更为优异，适用于一些需要更高灵敏度和更强动态行为的应用。但是两者都不是满映射系统，这直接导致了他们的参数的选取范围收到了限制。

如图 3.2 和图 3.3 所示：Logistic 系统在当 $3.5699 < \mu \leq 4$ 时，系统处于混沌状态。Tent 系统在当 $3.7500 < r \leq 4$ 时，系统处于混沌状态。本章介绍的加密系统需要参数当作密钥参数，Logistic 系统和 Tent 的参数选取范围使得加密系统的密钥空间变小，不利于系统安全性和复杂性。

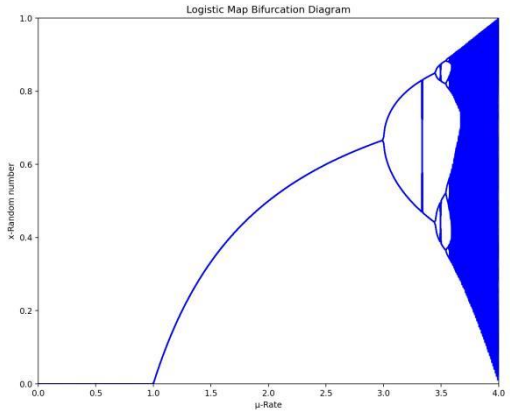


图 3.2 Logistic 系统分岔图

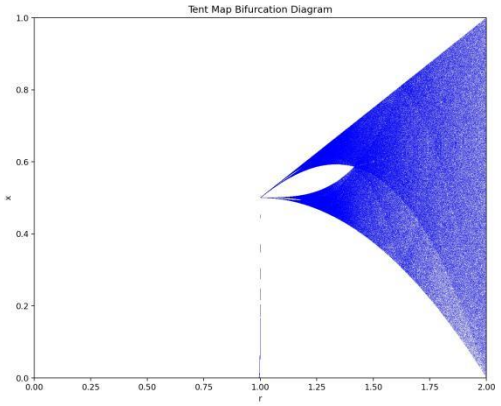


图 3.3 Tent 系统分岔图

本文采用 Logistic-Tent 系统，Logistic-Tent 系统的优点在于它结合了两者的特点，既保留了 Logistic 系统的简单性和稳定性，又引入了 Tent 系统的高灵敏度和多样性，使得在处理一些具有较高复杂度的混沌问题时更加高效和灵活，解决了单一系统在稳定性与复杂性之间的权衡问题。

Logistic-Tent 映射表示为公式(3.1)：

$$x_{n+1}=\begin{cases}\left[\mu x_n\left(1-x_n\right)+\left(4-\mu\right)x_n/2\right]\bmod 1, & x_n<0.5;\\ \left[\mu x_n\left(1-x_n\right)+\left(4-\mu\right)\left(1-x_n\right)/2\right]\bmod 1, & x_n\geq 0.5.\end{cases} \tag{3.1}$$

其中$\mu\in(0,4]$ 。

Logistic-Tent 的分岔图 3.4 所示：Logistic-Tent 是满映射系统和 Logistic，Tent 系统相比，有更加广阔的密钥空间。对加密模型的安全性和复杂性有正向的提升。

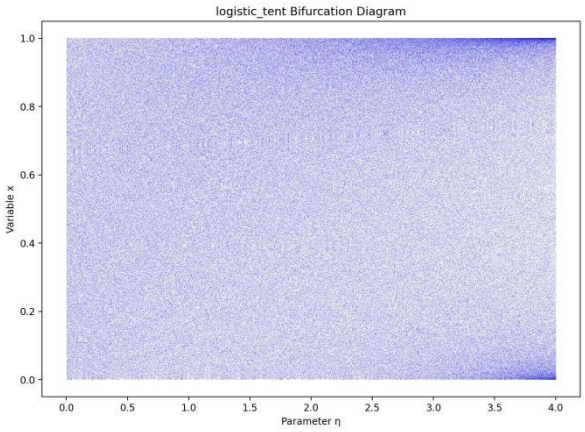


图 3.4 Logistic-Tent

3.2.2 三维混沌系统的选取

根据第二章内容，Chen 系统和 Lorenz 系统都属于经典的三维混沌系统，Chen 系统通常能比 Lorenz 系统产生更均匀、随机性更强的序列。更复杂的混沌行为使得系统对初始条件更加敏感，进而提升密钥敏感性和抗攻击性。

由图 3.5 可见，Chen 参数设置为 $a=35$ ，$b=3$ ，$c=28$，经过长时间运行后，系统只在三维空间的一个有限区域内运动，系统在此区域中的运动是混沌状态。

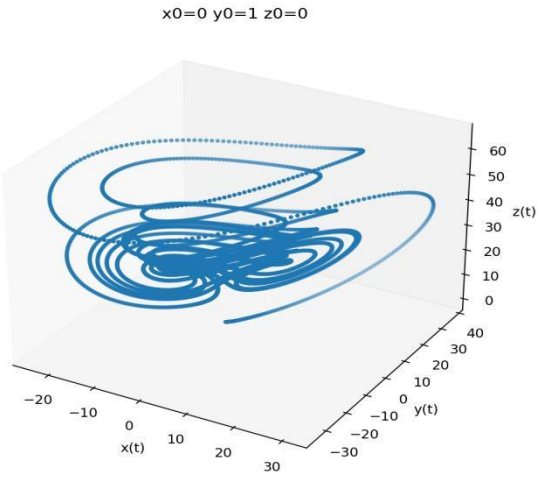


图 3.5 Chen 系统

从两个靠的很近的初值条件出发（y 只相差 0.0001）给出了 x(t)轨道的演化图 3.6 如下：橙色线条的 y 初始值位 1，蓝色线条的 y 初始值为 1.0001。

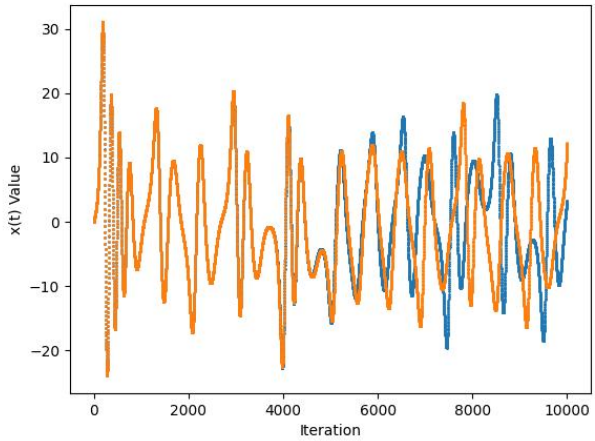


图 3.6 Chen 系统迭代图

由图 3.6 可见，随着时间的演化，可以看到原本靠得很近的轨道，在迭代 7000 次左右后 x 的值开始出现较大的区分，最后两条轨道变得毫无关联。

因为 Chen 是个三维的混沌系统，所以可以用 Chen 系统的三维混沌序列来操控 DNA 编码扩散情景下的 DNA 编码，解码操作和 DNA 运算操作操作。用 Chen 系统作为 DNA 编码扩散操作的混沌系统可以增加加密系统的混沌性。

3.3 混沌序列的生成方式

在双端协同存证模型中，唯一标识码作为原始图片的数字指纹，同时是置乱密钥的生成依赖，需满足全局唯一性、抗篡改性与高效生成需求。混沌序列作为加密载体 QR 码的动态控制参数，需要完成对 QR 码的置乱和扩散，必须具备强随机性和初值敏感性。

本节下采用一种元数据驱动的“标识-序列协同生成机制”，通过哈希函数确保标识码唯一性，结合一维和三维混沌系统生成动态密钥和多维混沌序列。因为每张图片的元数据不同，生成的散列值不同，置乱密钥也不同所以可以实现“一图一密”，确保载密 QR 码的信息安全。

3.3.1 生成唯一标识码

NIST 标准化哈希算法历经多代迭代，形成 SHA-0、SHA-1、SHA-2、SHA-3 四大分支。其中 SHA-2 系列包含六种子类（SHA-224/256/384/512/512-224/512-256），其核心优势体现在长哈希值设计（224-512 位）与抗碰撞强度提升，相较 SHA-1 与 MD5 具备显著安全性优势（SHA-1 碰撞攻击复杂度 2^63 次，SHA-256 达 2^128 次）。尽管 SHA-512 通过增加迭代轮数（80 vs 64）进一步强化安全性，但由此产生的计算开销导致吞吐率不如 SHA-256。综合考量安全性基线（满足 128 位抗碰撞）、运行效率（单位

时间处理量）及软硬件兼容性（广泛支持 AES-NI 指令加速），SHA-256 成为标识码生成函数的优化选择。使用 SHA-256 算法的算法生成唯一标识主要步骤如下：

（1）元数据转换：输入的元数据包括，用户编号，图像名称，拍摄设备，拍摄地点，文件提交时间，文件大小，文件类型，图片像素长度，文件像素宽度，文件创建时间，本地存证生成 32 位随机英文和数字编码，将这些数据都按照 unicode 编码，然后将编码乱排，将乱排结果作为散列函数的输入。

（2）比特填充：对于输入的元数据字符串，其长度为 L，需要在消息的末尾添加填充比特。填充的具体方法如下：首先，在消息末尾添加一个 1 位的比特。接着填充 K 个 0，其中 K 是满足方程 L+K+1=448mod512 的最小非负整数。最后，附加消息原长度 L 的二进制表示。经过这种填充后，最终的消息长度将是 512 的整数倍。

填充比特的具体规则请参见图 3.7。为了进一步说明这一过程，假设输入比特串为 "a, b, c"。

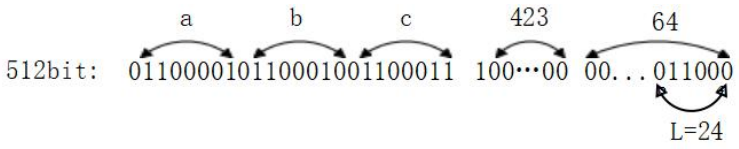


图 3.7 比特填充

（3）第三步是算法输出的初始化：在每次执行 SHA-256 计算时，首先需要进行输出初始化。该过程使用 8 个 32 位的寄存器来保存 SHA-256 在每个计算步骤中的中间结果。根据算法的协议标准，初始化值由前 8 个质数平方根的小数部分的前 32 位组成。这些初始值可以通过 16 进制表示如下：

$$H_0^0=0x8a46e667$$
$$H_1^0=0xbb67ae85$$
$$H_2^0=0x3c6ef372$$
$$H_3^0=0xa54ff53a$$
$$H_4^0=0x510e527f$$
$$H_5^0=0x9b05688c$$
$$H_6^0=0x1f83d9ab$$
$$H_7^0=0x5be0cd19$$

（4）常量数组的初始化：该数组中的值来自前 64 个质数（从 2 到 311）对应立方根的小数部分的前 32 位。所有常量的 16 进制表示形式按顺序排列，如表 3.2 所示。

表 3.2 常量数值表

常量数值表							
428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d698aa4a	f40e3585	106aa070
19a4c116	1e376c08	2748776c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506cbe	bef9a3f7	c67178f2

（5）数组w_i的计算，w_0~w_{15}的值分别由 512 比特输入块按 32 比特从高到低分割得到，数组中的w_{16}~w_{63}的获取方式如式(3.2)所示：

$$w_i = w_{i-16} + w_{i-7} + s_0 + s_1 \tag{3.2}$$

式中的参数s_0 ，s_1的获取方式如式(3.3)和(3.4)所示：

$$s_1 = (w_{i-15} \gg 7) \oplus (w_{i-15} \gg 18) \oplus (w_{i-15} \to 3) \tag{3.3}$$

$$s_1 = (w_{i-2} \gg 17) \oplus (w_{i-2} \gg 19) \oplus (w_{i-2} \to 10) \tag{3.4}$$

压缩函数的迭代计算：SHA-256 算法进行 64 次迭代计算。在每次迭代中，8 个常数变量 a, b, c, d, e, f, g, h 都是 32 位的数据变量。首先，需要初始化这 8 个迭代常数变量，其初始化公式如式(3.5)所示：

$$\begin{aligned} a &= H_0^{(i-1)} \\ b &= H_1^{(i-1)} \\ c &= H_2^{(i-1)} \\ d &= H_3^{(i-1)} \\ e &= H_4^{(i-1)} \\ f &= H_5^{(i-1)} \\ g &= H_6^{(i-1)} \\ h &= H_7^{(i-1)} \end{aligned} \tag{3.5}$$

接下来，需要对这八个常数进行迭代更新。八个常数的迭代公式如下式(3.6)所示：

$$\begin{aligned} T_1^t &= h_{t-1} + \sum_1^{\{256\}}(e_{t-1}) + CH\left(e_{t-1}, f_{t-1}, g_{t-1}\right) + K_t^{\{256\}} + W_t \\ T_2^t &= \sum_0^{\{256\}}(a_{t-1}) + Maj(a_{t-1}, b_{t-1} + c_{t-1}) \\ a_t &= T_1^{(t-1)} + T_2^{(t-1)} \\ b_t &= a_{t-1} \\ c_t &= b_{t-1} \\ d_t &= c_{t-1} \\ e_t &= d_{t-1} + T_1^{t-1} \\ f_t &= e_{t-1} \\ g_t &= f_{t-1} \\ h_t &= g_{t-1} \end{aligned} \tag{3.6}$$

其中函数 CH，Maj，$\sum_0^{\{256\}}(x)$，$\sum_1^{\{256\}}(x)$的运算公式如下所示：

$$CH(x,y,z) = (x \,\&\, y) \oplus (\neg\, x \,\&\, z) \tag{3.7}$$

$$Maj(x,y,z) = (x \,\&\, y) \oplus (x \,\&\, z) \oplus (y \,\&\, z) \tag{3.8}$$

$$\sum_0^{\{256\}}(x) = (x \gg 2) \oplus (x \gg 13) \oplus (x \gg 22) \tag{3.9}$$

$$\sum_1^{\{256\}}(x) = (x \gg 6) \oplus (x \gg 11) \oplus (x \gg 25) \tag{3.10}$$

最后，计算每一步的 Hash 值时，需要使用上一步（i-1）的 Hash 值以及更新后的常数变量。具体的计算方法如式(3.11)所示：

$$\begin{aligned}
H_0^{(i)} &= a+H_0^{(i-1)} \\
H_1^{(i)} &= b+H_1^{(i-1)} \\
H_2^{(i)} &= c+H_2^{(i-1)} \\
H_3^{(i)} &= d+H_3^{(i-1)} \\
H_4^{(i)} &= e+H_4^{(i-1)} \\
H_5^{(i)} &= f+H_5^{(i-1)} \\
H_6^{(i)} &= g+H_6^{(i-1)} \\
H_7^{(i)} &= h+H_7^{(i-1)}
\end{aligned} \tag{3.11}$$

在处理完最后一个 512 比特的消息块后，最终的 SHA-256 输出将是最后一次迭代的运算结果。该结果的输出形式如公式(3.12)所示：

$$H=H0|H1|H2|H3|H4|H5|H6|H7 \tag{3.12}$$

在上面各个公式中，>>表示的是循环右移，→标识的是右移操作，符号|表示的是位的拼接，符号⊕表示的是异或，┐表示的是取反运算，&表示的是按位与运算。

3.3.1 生成混沌序列

（1）一维置乱混沌矩阵的生成过程

在上一节中，使用 SHA-256 哈希算法对元数据进行了处理，从而生成了相应的哈希值 H。这一节中，将十六进制格式表示哈希值转换为一个由 256 位二进制数字组成的唯一标识码 key。随后，将 key 按每 32 位一组进行划分，将每组二进制数转换为十进制大数，并通过逐项相加，再通过式(3.13)进行计算，得到混沌序列的混沌初值。SHA-256 哈希算法的一个重要优点是，输出序列与明文之间有着紧密的关联，因此，当元数据发生变化时，生成的哈希值也会随之不同。又由于每个图片的元数据都不同，所以每张图片生成的哈希值不同，因此每张图片的根据唯一标志码生成的置乱密钥都不同，做到了一图一密，增加了加密模型的安全性。

$$key=\begin{cases}k_1=\{k_1,k_2,...,k_{32}\}\\k_2=\{k_{33},k_{34},...,k_{64}\}\\k_3=\{k_{65},k_{66},...,k_{96}\}\\k_4=\{k_{97},k_{98},...,k_{128}\}\\k_5=\{k_{129},k_{130},...,k_{160}\}\\k_6=\{k_{161},k_{162},...,k_{192}\}\\k_7=\{k_{193},k_{194},...,k_{224}\}\\k_8=\{k_{225},k_{226},...,k_{256}\}\end{cases} \tag{3.13}$$

接下来根据上文得到的 8 个大数字，根据式(3.14)生成一维混沌序列的混沌初值 x_0。x_0 就是 logistic-tent 混沌映射其中一个置乱密钥，也是混沌映射系统的混沌初值。

$$x_0=(\sum_{i=1}^{8}\mathrm{k}_i)*10^{-12} \tag{3.14}$$

logistic-tent 混沌映射还有一个置乱参数 $\mu\in(0,4]$，那么 μ 的生成方式如下式(3.15)所示，参数由用户提供存证产生精度位 1e-8 的 0 到 4 之间的小数。

$$\mu=random(0,4) \tag{3.15}$$

由此可以得到一个值在(0, 4]之间的混沌参数 μ，这是一维混沌序列的第二个置乱密钥。

带入初始值 x_0 和 μ 迭到式(3.1)，迭代生成和 M×N（M 和 N 分别是生成二维码的厂像素长度和像素宽度）长度的序列 S。

（2）三维混沌矩阵的生成过程

三维序列的 Chen 系统选取的参数是 $a=35$ ，$b=3$ ，$\mathrm{c}=28$，在 3.2.2 节看到了，在这个参数体系下，系统呈现出良好的混沌性以及初值敏感性。那么只要确定 x，y，z 三个初值就可以通过迭代获得三维的混沌序列了。首先可以将求一维混沌序列的值 x_0 和 μ 作为 x 和 y 的初始值。z 的初始值如式(3.16)所示：

$$z_0=-20+\mathrm{mod}((\sum_{i=1}^{8}\mathrm{k}_i),40) \tag{3.16}$$

其中，-20 的目的是将 z 的初始值控制在-20 到 20 之间。所以综上所述，Chen 系统的初始值 u_0 如式(3.17)所示：

$$u_0 = [x_0, \mu, z_0] \tag{3.17}$$

假设图像大小为 M×N，系统生成 10000+M×N 长度的伪随机序列，并丢弃前 10000 个值以获得更好的随机效果，得到 M×N 长度的混沌序列 X ，Y ，Z。

$$\begin{aligned} \mathbf{X} &= \{x_1, x_2, ..., x_{MN}\} \\ \mathbf{Y} &= \{y_1, y_2, ..., y_{MN}\} \\ Z &= \{z_1, z_2, ..., z_{MN}\} \end{aligned} \tag{3.18}$$

混沌序列 X,Y,Z 就是三维的混沌序列，用于 DNA 编码扩散和 DNA 运算控制。

3.4 QR 码的置乱与扩散

在上一节介绍了一维混沌序列和三维混沌序列的产生步骤。有了混沌序列之后就可以对 QR 码进行混沌置乱和扩散的操作。这一届主要介绍 QR 码的混沌置乱的具体步骤：

第一步：利用唯一标识码生成唯一的 QR 码

第二步：将洗牌算法和 logistics-tent 混沌映射系统生成的混沌序列结合进行像素的置乱。

第三步：将 DNA 编码运算和 Chen 系统的混沌序列结合，将第二部生成的置乱图像进一步像素扩散。

3.4.1 QR 码的生成

根据生成的散列值 H，进一步生成相应的二维码。现如今，互联网上存在多种二维码生成工具，但是他们定制化能力很低，因为后面章节要对二维码进行分块加密，所以对二维码边长有一定要求，而且后面章节需要对二维码序列化所以基于开源的 python 项目 qrcode，开发了一个二维码生成软件，能够将文本信息转换为二维码并且可以将二维码序列化为 0，1 序列。

下来展示的是通过自定义的二维码生成软件生成的二维码效果：二维码内容：“hello”由 SHA256 算法得出的散列：

图 3.8 hello 的散列二维码

由以上二维码对比可以看出，sha256 算法生成的 256 位的散列值用 16 进制转换后的二维码复杂度适中。

3.4.2 QR 码的置乱操作

置乱操作的目的式将一个序列打乱，但是什么才是真正的乱？对于包含有 n 个元素的序列，由于这个序列的排列方式有 n!种，所以意味着，将这个元素完全打乱后有 n!种可能，如果序列组足够混乱则产生的每一种排列的可能都是相同的。以上是从序列整体来分析。对于序列中的每一个元素来说，如果序列足够的混乱，则某个元素出现在序列中任何位置的可能性都是相同的，即任何一个元素，出现在任意一个位置的概率都是 1/n。

Fisher-Yates 洗牌算法已经在第二章详细介绍，Fisher-Yates 算法的核心思想是逐步缩小随机选择的范围，确保每个位置的元素仅与未固定的位置交换。但是该算法的时间复杂度为 O(n^2),空间复杂度为 O(n)。所以选择经过优化后的洗牌算法，Knuth-Durstenfeld 算法，该算法不需要删除元素和额外空间，时间复杂度为 O(n),空间复杂度为 O(1)，Knuth-Durstenfeld 算法的流程：

1.输入：长度为 n 的数组 A = [$a_0, a_2, \ldots a_{n-1}$]

2.遍历方向：从后向前遍历，索引从 n-1 到 0

3.步骤：

(1)对于每个位置 i（从 n-1 到 0），生成随机整数 j∈[0,i]

(2)交换 A[i]和 A[j]

(3)循环步骤(1)(2)指导到达第一个位置

4.输出：一个完全打乱的数组

由上述的算法步骤可以看出 Knuth-Durstenfeld 算法整体非常简单，我们可以很容易发现，算法中的“随机生成整数”步骤非常关键，因为这直接决定了洗牌算法的随机性。这里可以将洗牌算法和上一节生成的混沌序列结合。用混沌序列的值代替随机函数生成的 j∈[0,i]。步骤如下：

步骤 1：将元数据生成的二维码图像信息转变成比特流序列 J。

步骤 2：利用密钥 x_0 和 μ，结合 logistic-tent 混沌产生与比特流长短相等的混沌序列 S（上一节已经详细介绍）。

步骤 3：从比特流序列 P 的最后一位像素开始循环

步骤 4：每次循环的像素位置为 i，每次循环 i 都减小一位。

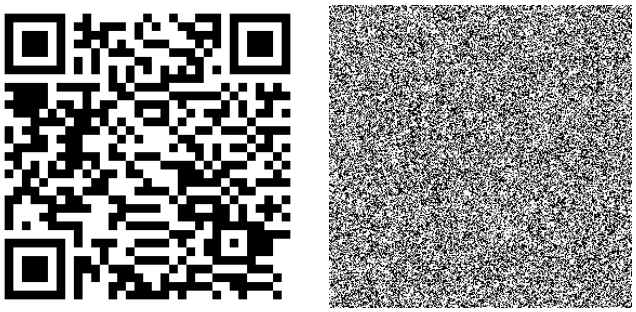
步骤 5：利用混沌序列 S 生成随机下标 j，如式(3.19)所示。down()函数表示的是向下取整操作。

$$j=down(i*S(i)) \tag{3.19}$$

步骤 6：交换 J(i)和 J(j)

步骤 7：一直重复步骤 3 到步骤 6 到第一个元素，就得到置乱后的图像 P。

经过上面的步骤就可以得到了置乱后的 QR 码，用 Knuth-Durstenfeld 算法来保证理论上置乱的混乱性，又用 logistic-tent 混沌体系产生的混沌系统来保证算法中最关键的随机性。下面是经过置乱前的 QR 码和置乱后的 QR 码的对比。



QR 码　　置乱后的 QR 码

图 3.9 hello 的散列二维码

3.4.3 QR 码的扩散操作

上一节已经对二维码进行了置乱操作，置乱操作通过改变像素的位置和排列方式，破坏图像的空间相关性。但是这样的置乱并没有改变像素原有的值，原始图片只是发生了像素的位置变化，并没有发生像素值的变化。这样置乱图片的统计学特征并没有和原始图像产生区别，比如置乱图像的直方图，像素值分布等。如果攻击者获得了部分的明文信息可能通过对比来推测置乱的规则，也可以利用统计学特征来还原信息。

本节的扩散操作就是要改变置乱图像的像素值，目的是通过扩散操作使像素值的分布趋于平均，这样直方图的分布就平坦化，隐藏了原始图像的统计学特征。而且扩散操作需要扩散密钥，扩散密钥和置乱密钥结合，使得攻击者及时破解了其中一个密钥也不

能复原原始图像的内容。下面将详细介绍基于 Chen 系统的 DNA 编码扩散操作。

DNA 编码扩散操作需要确定原始图像和扩散矩阵的编码规则，原始图像和扩散矩阵的 DNA 运算规则，还有 DNA 解码规则。以上三种规则的选择会影响最终的扩散效果，本文将混沌序列和 DNA 编码扩散结合，用混沌序列的无序性来选择 DNA 编码扩散步骤中的各种规则选择。下面是 DNA 编码扩散算法的实现。

步骤 1：将混沌序列 X 的元素值映射到[0,1]内，将映射完毕的矩阵当作一个扩散矩阵。如式(3.20)所示，shape()函数是将混沌序列构造成一个 $N\times N$ 的矩阵 C，N 是二维码的边长。式(3.21)表示的是矩阵 C 每个元素的值的确定过程，i∈[1, $N\times N$]，函数 down()是向下进行取整操作，这么做的目的是将混沌序列转成成一个 $N\times N$ 的二值矩阵。对 N 要求是 8 的倍数，因为我们生成的二维码可以调整边长，所以可以保证 N 是 8 的倍数。

$$\mathrm{C}=\mathrm{shape}(M,N) \tag{3.20}$$

$$\mathrm{C}_i=\mathrm{mod}(\mathrm{down}(x_i\times 10^8),2) \tag{3.21}$$

步骤 3：开始对 C 和P进行编码操作：将 C 和P分成 4×4 的像素块，这样可以用多线程编码来提高 DNA 编解码的速度，同时将步骤 1 产生的矩阵 C 也分成 4×4 的矩阵。需要对这两个矩阵进行 DNA 编码，对矩阵的 DNA 编码方式由式(3.22)决定。

$$w_{1i}=\mathrm{mod}(\mathrm{C}^{'}_i,8)+1 \tag{3.22}$$

其中w_i表示置乱图像和扩散矩阵各块的编码方式，w_i得到的数字分别对应着第二章给出的 DNA 编码的 8 种编码方式，在第二章已经给出了每个号码对应的编码方式。

接下来要确定矩阵$P^{'}$和扩散矩阵$C^{'}$之间的 DNA 编码的运算规则了，在第二章种介绍了 DNA 编码的运算规则有三个，分别是加法、减法、异或和同或。可以用数字 0 到 3 来代表四种 DNA 的运算规则。通过式(3.23)和混沌矩阵 Y 配合，确定了原始图像每一块区域和混沌矩阵 C。

$$w_{2i}=\mathrm{mod}(\mathrm{round}(y_i\times 10^4),4) \tag{3.23}$$

最后要确定 DNA 的解码规则，通过式和混沌矩阵 Z 配合，可以获得解码的方式。同编码方式一样，DNA 解码方式有 8 种。

$$w_{3i}=\mathrm{mod}(\mathrm{round}(z_i\times 10^4),8)+1 \tag{3.24}$$

DNA 扩散过程如图 4 所示。

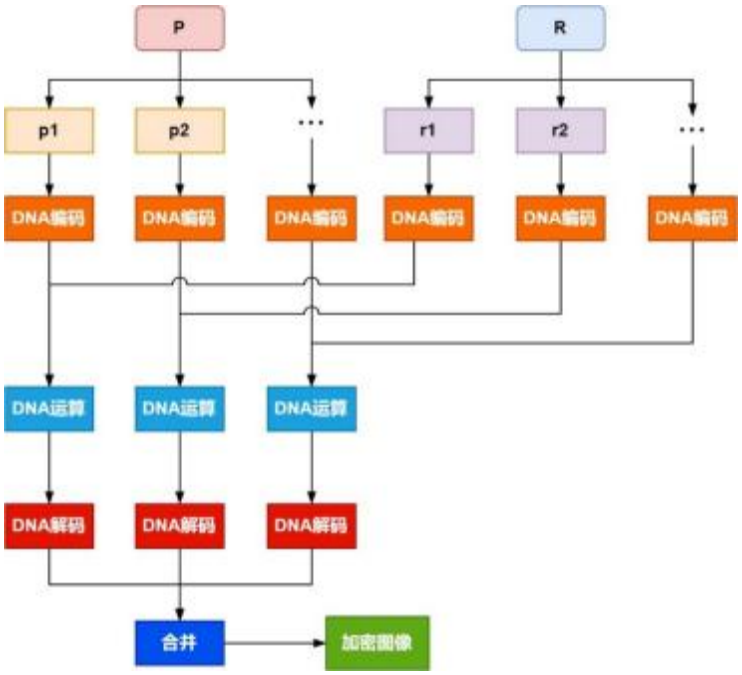


图 3.10 DNA 扩散过程

将扩散矩阵 C 和图像矩阵 P 执行压缩操作后，再将矩阵 C' 和 P' 分成块矩阵，对分块后的矩阵依次进行 DNA 编码以及运算，再将运算后的结果进行 DNA 解码，所获的图像像素值与原始图像将完全不相同。合并后即可获得加密图像。

3.5 实验分析

本节会就前文的置乱-扩散算法展开实验分析：对算法的安全性和有效性进行测试。本节将采用一些 280×280 大小的二值图片进行如下测试：图像复原实验、密钥敏感性实验、直方图分析、图像像素相关性分析、抗噪声盛宴、抗裁切实验以及抗差分攻击分析。

3.5.1 加密图像复原

本实验将置乱-扩散后的图像进行加密过程的逆过程，恢复原始图片。然后测试是否能正常扫码出原始信息，并且将进行图像的相似度测试。查看复原出的二维码图片是否和原始图片一模一样。

下面简单介绍图像的解密流程：

步骤一：获取载密图像 P_s，密钥 H，密钥 μ。通过 H 和 μ 生成 logistic-tent 混沌系统和 Chen 系统的混沌初值。

步骤二：根据混沌初值迭代生成一维混沌序列 S 和三维混沌序列 X，Y，Z。

步骤三：执行 DNA 编码扩散的逆操作。依靠序列 X，Y，Z，进行编码，运算，解码操作，共计 6 轮得到置乱图像。

步骤四：依赖步骤三得到的置乱图像和序列 S，执行 Knuth-Durstenfeld 洗牌算法的逆流程，得到原始图像。

图 3.11 是由置乱-扩散图像到置乱图像在到原始图像的解密效果。

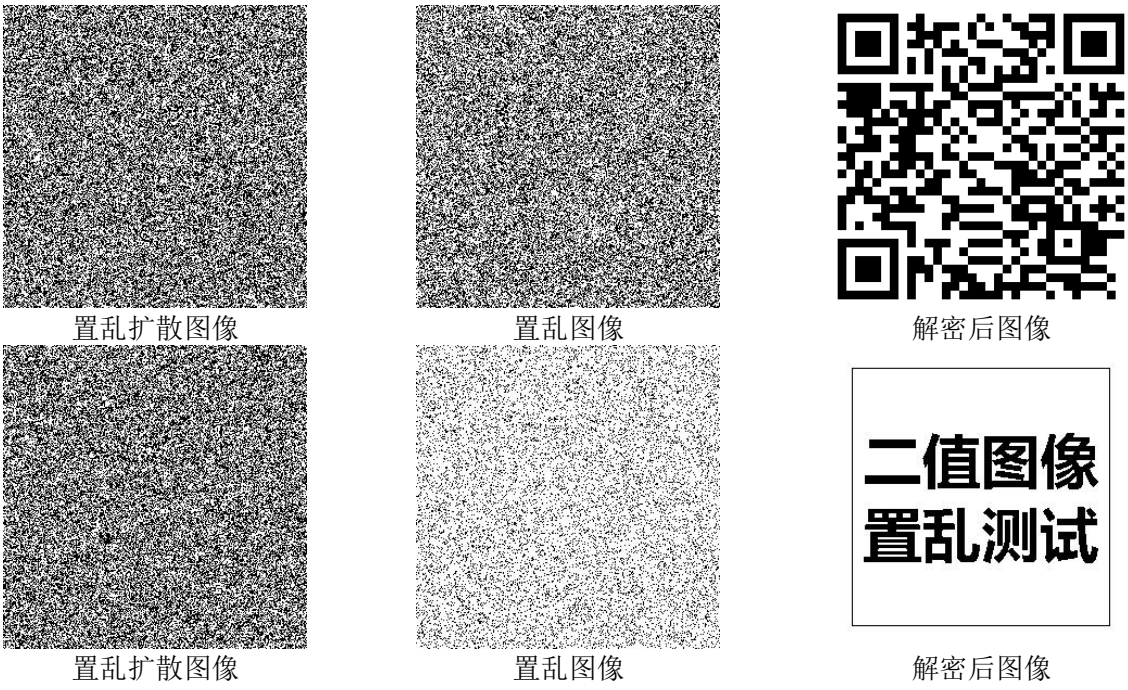


图 3.11 复原图像对比

通过扫码可以复原原始图像的二维码信息，通过观察法可以看到文字信息也能完全复原。我们还可用均方误差来像素级的查看图像是否被精准还原。式就是均方误差的公式。

$$MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[P_{\mathrm{a}}(i,j)-P_{b}(i,j)\right]^{2} \tag{3.25}$$

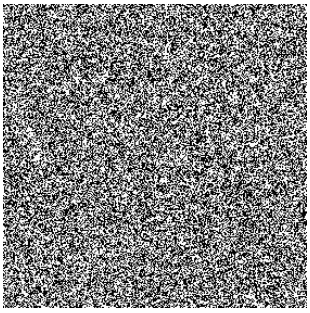
其中P_a和P_b分别代表着原始图像和解密后图像。m 和 n 是要对比的两个图像的像素高度。通过计算，解密后的图像和原始图像的均方误差是 0，这证明了加密过程的可逆性。该加密解密过程可以无损的处理二维码图像。

3.5.2 密钥敏感性测试

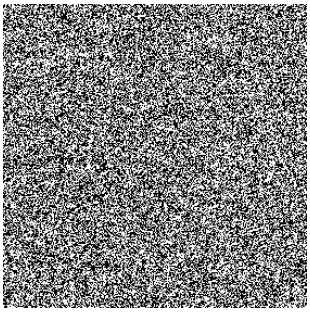
本文采用的密钥有两个：一个是将原始图像的元数据和图像哈希值当作输入参数，

经过 SHA256 算法运算得到的 256 比特的二进制数据；二是用户输入的(0，4]之间的，精度 1e-8 的小数，所以密钥空间大小为$2^{256}\times4\times10^{9}$，当密钥空间大于$2^{100}$的时候会有效的抵挡暴力破解。现在测试将 256 位的二进制的密钥中的最后一位反转，另一个密钥保持不变。看一下在只改变一位的情况下能否复原出原始图像。

原始图像



置乱扩散图像

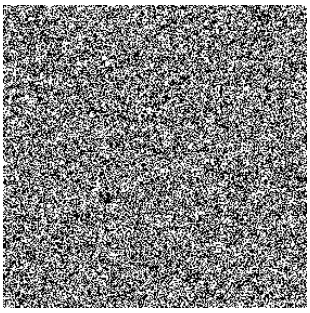


复原后图像

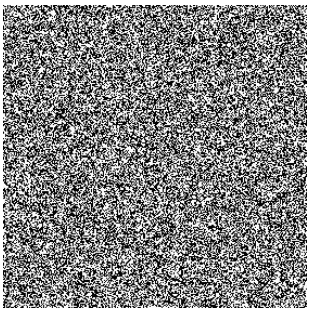
图 3.12 改变密钥解密图像

如图 3.12 所示，在密钥一被轻微修改之后复原出来的图片和原始图片完全不相同。接下来测试密钥二，最后一位小数加一由 1.68754124 变成 1.68754125 后的复原效果.

原始图像



置乱扩散图像



复原后图像

图 3.13 改变密钥解密图像

可以看从图 3.12 和图 3.13 看到，当无论是密钥一还是密钥二发生一点扰动，最后解密的图像都是完全混沌不可读的无用信息。

3.5.3 抗干扰测试

加密的图片在传播过程中会遇到一些干扰，比如加密图片被污损了一部分，造成了一部分图像区域不可见。也有可能传输中有干扰，图片中有一些噪声。所以本文进行抗剪切测试和抗椒盐噪声干扰测试。

1. 抗剪切测试

抗剪切测试是测试加密图像有一部分不可读，还能否复原出原始图像，复原出的图

像可读性会受到多大的影响，以此来测试加密算法的鲁棒性。本次实验将剪切部分的图像全部变成黑色。

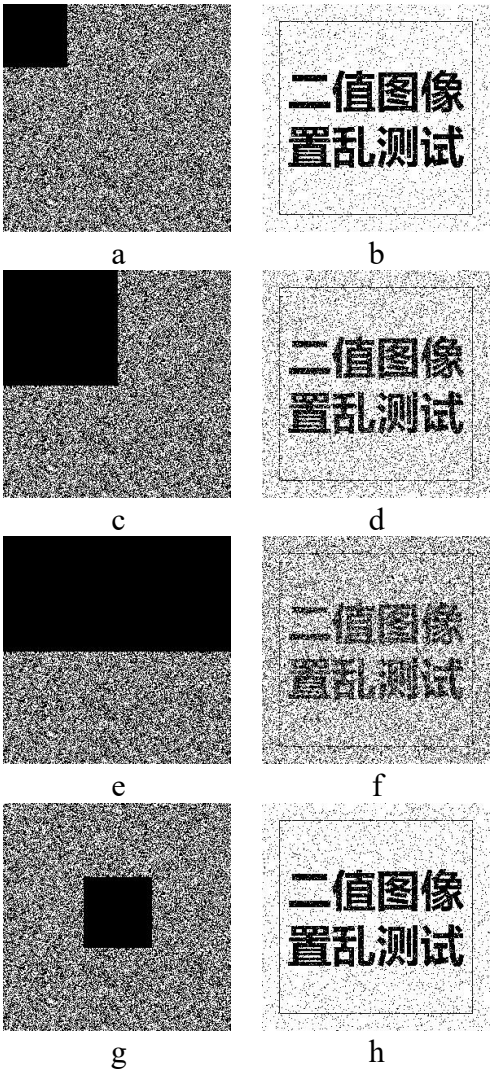


图 3.14 抗剪切能力测试

从图 3.14 可以看出来，随着剪切的面积增大图像的噪声逐渐增多，但是即使是有二分之一的面积被剪切之后，依然能识别出水印。

2. 抗椒盐噪声测试

抗椒盐噪测试是测试加密图像中因为传输质量原因，密文中被混入了噪声，测试还能否复原出原始图像，复原出的图像可读性会受到多大的影响，以此来测试加密算法的鲁棒性。图中测试了 3 中不同程度的噪声干扰：a，b，c 中分别添加了强度 0.1，0.2，0.3 的噪声。可以看图 3.15 是复原后的原始图像，还原图像内容依然可见。

0.1 噪声　　0.2 噪声　　0.3 噪声

图 3.15 抗剪切能力测试

除了肉眼观察图像，还可以用峰值信噪比（Peak Signal to Noise Ratio，PSNR）来量化系统抵抗干扰的能力。PSNR 用来衡量原始图像与处理过后的图像的差异。表 3.2 可以看出虽然随着干扰的程度越高，PSNR 的值降低，但是依然在可以辨认的范围内，仍然可以辨认出原始图像的有效信息。峰值信噪比的公式(3.26)如所示：

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \quad [\mathrm{dB}] \tag{3.26}$$

其中 MAX 表示信号中像素值的最大可能取值。对于 8 位图像，MAX 通常为 255（即2^8-1）。MSE（Mean Squared Error，均方误差）为原始信号与处理后信号对应像素差的平方的平均值，计算公式前面已经给出，为式(3.25)：

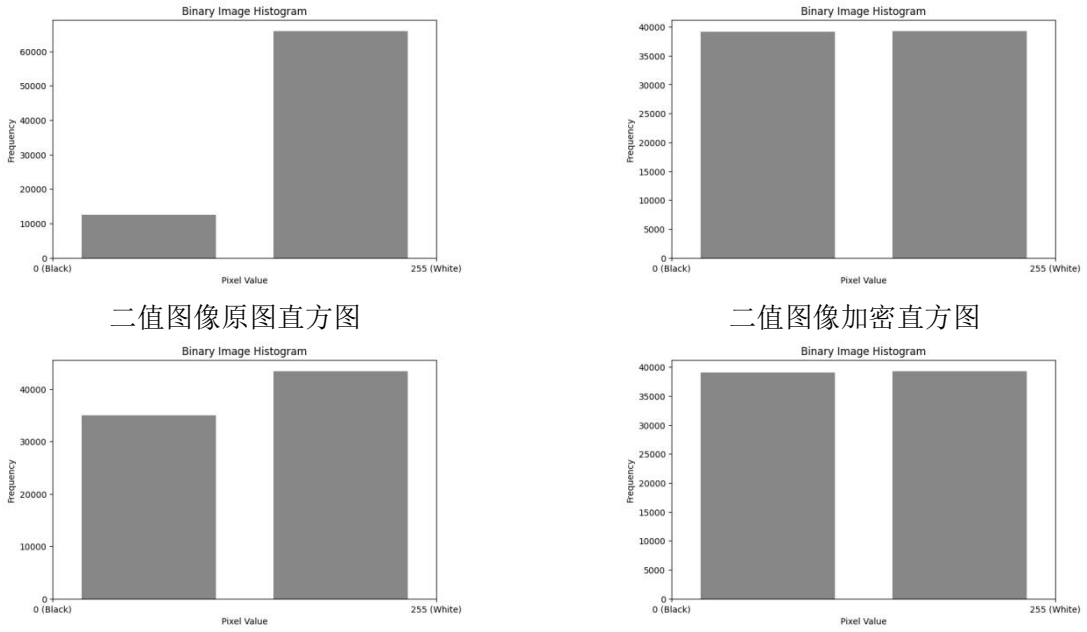
表 3.3 不同干扰下的 PSNR 值

测试项目	测试强度	PSNR(dB)
剪切测试	左上角 1/16 剪切	16.5651
剪切测试	左上角 1/4 剪切	12.6589
剪切测试	上方 1/2 剪切	8.2657
剪切测试	中间 1/8 剪切	16.2487
椒盐噪声测试	0.1 强度椒盐噪声	15.2368
椒盐噪声测试	0.2 强度椒盐噪声	13.5687
椒盐噪声测试	0.3 强度椒盐噪声	10.9874

3.5.4 直方图测试

直方图可以量化的的看到图像的颜色分量或者灰度等级的分布情况。如果加密模型的置乱和扩散状况良好，那么原始图片和加密图片的像素值分布表现不同，加密图像像素分布更加平均，呈现一种均匀分布的状态，所以不会体现原始图片的像素分布特征。

对二维码和二值图像侧视图及他们的加密图像进行直方图分布，如下图 3.16 所示：由于是二值图像，图像只有 0 和 255 两种像素。可以看到在进行加密后，黑白像素的比例趋近于 1：1.加密模型很好的隐藏了原始图像的像素分布。



二值图像原图直方图　　二值图像加密直方图

二维码原图直方图　　二维码加密直方图

图 3.16 直方图对比

3.6 本章小节

本章提出了一种结合了混沌置乱扩散的二值图像加密方案。该方案首先依据用户上传的图像哈希值和图像元数据生成和该图像绑定的哈希值作为加密方案的其中一个加密密钥，用户输入的混沌系统的参数作为加密方案的第二个密钥。将这两个密钥用于 logistic-tent 混沌系统和 Chen 混沌系统的系统初值生成，进一步生成一维置乱序列和三维置乱序列。用于对二值图像的置乱和扩散操作。

在实行混沌置乱和混沌扩散的步骤时，加密方案将 Knuth-Durstenfeld 洗牌算法和 logistic-tent 混沌系统结合，将二值秘密图像像素重新排序。像素重排完毕后进行扩散操作，加密方案将 DNA 编码扩散和 Chen 混沌系统结合，用 Chen 系统生成的三维混沌序列中的 X 混沌序列来充当扩散矩阵和选择编码方式，Y 序列决定每次 DNA 运算的运算规则，Z 序列决定每次解码操作的解码规则。从而达到扩散运算的扩散矩阵随机、编码随机、运算随机和解码随机。完成了置乱和扩散两步后生成秘密图像。

最后，对生成的秘密图像进行实验测试，包括复原实验，密钥敏感性实验，剪切实验，噪声实验，直方图对比。经过测试证明加密模型有很好的无损加密解密，一定的抗

攻击和较强的鲁棒性。

第四章　DCT 域数字水印嵌入算法

按照第三章的方法，将原始图像的特征信息生成了二维码，并且将二维码经过了置乱与扩散处理。得到了一张无序混乱的二值图像。第四章要做的就是将该二值图像当作数字水印隐藏到原始图像中去，并且嵌入算法要兼顾鲁棒性、不可见性和安全性。基于以上要求，本文提出结合 HVS 和混沌系统的 DCT 域的二值图像数字水印嵌入算法，该算法将载体图像分块分析，量化的分析图像每块不同区域的是否适合嵌入信息，优先给适合嵌入信息的图像块嵌入更多信息，然后根据混沌序列将秘密信息完全随机的嵌入到不同图像块 DCT 域中去，最后通过逆 DCT 变换恢复原始灰度图，加上原始载体图像的颜色通道，完成嵌入流程。最后通过加入高斯噪声、椒盐噪声以及 JPEG 压缩和剪切等处理操作，结果表明了该算法具有很好的视觉掩蔽特性和鲁棒性。

4.1 算法模型介绍

中在介绍 DCT 域嵌入算法前，我们已经做了一些准备工作，包括：

1.对二维码信息进行混沌置乱；

2.对混沌置乱后的水印信息进行混沌扩散，实现二次加密；

完成准备工作后，需要将第三章生成的二值数字水印嵌入到载体图像中。嵌入过程使用本章设计的 DCT 域的数字水印算法，算法的大致流程如下：

步骤 1：将载体图像转换成灰度图像，按照 8×8 分块。

步骤 2：计算每一块图像的均值、方差、熵值，并给图像块打分，分数越高，越适合隐写数据，相应的嵌入强度越高。

步骤 3：根据混沌系统确定嵌入随机数序列。

步骤 4：根据混沌系统确定每个比特要嵌入的 DCT 系数位置。

步骤 5：将秘密信息按照步骤 4 选择的插入位置

步骤 6：重复步骤 4，5，直到所有的秘密元素都被隐藏到载体图像中。

步骤 7：将嵌入完成的载体图像恢复 RGB 通道色彩。完成整个嵌入流程。

4.2 离散余弦变换介绍

4.2.1　DCT 变换的原理

DCT 的全称是离散余弦变换(Discrete Cosine Transform)，DCT 可以将空域上的信号映射到频域上。从而可以让我们将对信号的研究和操作从空域空间转到了频域空间。由于 DCT 域的水印算法发生在频域空间，他的水印嵌入和空域常用的 LSB 算法比起来具

有更高的稳定性，对常见的攻击有优秀的鲁棒性，同时 DCT 水印技术具有良好的可逆性，所以我们可以靠 DCT 的逆变换将频域信息重新转换成图像信息，并且对图像几乎无损。

DCT 可以将一维或者二维的离散序列转换成一组余弦函数的系数序列。这组余弦函数表示了原始数据中的频谱特征。基于以上的原理，我们可以将图像的信号分解成一系列的不同频率成分的系数。式(4.1)就是一维的 DCT 的定义。

$$F(u)=C(u)\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\frac{(2x+1)u\pi}{2N} \tag{4.1}$$

在上式中，*f(x)*就是一维的信号序列，*C(u)*如式(4.2)所示：

$$C(u)=\begin{cases}\dfrac{1}{\sqrt{2}} & u=0\\ 1 & \text{其他}\end{cases} \tag{4.2}$$

DCT 同时可以进行可逆变换，这也方便了我们对信号进行还原操作，DCT 的逆变换如式所示：

$$f\left(x\right)=\sqrt{\frac{2}{N}}\sum_{u=0}^{N-1}C(u)F(u)\cos\frac{(2x+1)u\pi}{2N} \tag{4.3}$$

我们发现了，DCT 无论正向变换还是逆向变换，都有式(4.4)在

$$g(u,x)=C(u)\sqrt{\frac{2}{N}}\cos\frac{(2x+1)u\pi}{2N} \tag{4.4}$$

我们将他提取出来可以得到式(4.5)。其中的 *M* 可以表示成为式(4.6)一个矩阵。

$$\mathrm{F=Mf} \tag{4.5}$$

$$\mathrm{M}=\begin{bmatrix} 1/\sqrt{N} & [& 1 & 1 & ... & 1 & 1 \\ \sqrt{2/N} & [& \cos\frac{\pi}{2N} & \cos\frac{3\pi}{2N} & ... & \cos\frac{(2N-1)\pi}{2N} &] \\ \sqrt{2/N} & [& \cos\frac{2\pi}{2N} & \cos\frac{6\pi}{2N} & ... & \cos\frac{(2N-1)2\pi}{2N} &] \\ ... & ... & ... & ... & ... & & \\ \sqrt{2/N} & [& \cos\frac{(N-1)\pi}{2N} & \cos\frac{3(N-1)\pi}{2N} & ... & \cos\frac{(2N-1)(N-1)\pi}{2N} &] \end{bmatrix} \tag{4.6}$$

了解到完一维的 DCT 变换，可以将定义推广到二维的 DCT 变换中，将之前的一维序列 *f(x)*推广到二维序列 *f(x,y)*。*f(x,y)*的 DCT 变换为式(4.7)：

$$F(u,v)=\frac{2}{\sqrt{MN}}C(u)C(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2N} \tag{4.7}$$

二维的 DCT 逆变换如式(4.8)所示：

$$f(x,y)=\frac{2}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2N} \tag{4.8}$$

4.2.2 DCT 的系数

一般进行 DCT 变换首先要做的就是对图像进行分块，一般选取的分块是 8×8 分块，意思是将原始图像分成 8×8 像素大小的小块，然后对每一块进行 DCT 变换，这样每一个小块都会得到 64 个 DCT 系数。为了使 DCT 系数矩阵能够呈现出规律性，会将系数矩阵进行 Zig-Zig 排序，如图 4.1 所示。

DCT 系数分为直流分量（DC 系数）和交流分量（AC 系数）。其中 DC 系数位于系数字水印的数矩阵的左上角(0,0)位置，他反映了图像块的平均亮度或能量，是其他所有像素值的加权平均。AC 系数则是除了 DC 系数外的其他系数，他反映了图像中不同频率的细节信息。DCT 变换后大部分的能量集中在低频区域，对应着图像的平滑区域，高频区域对应的是图像的边缘，纹理等细节。图 4.1 就是经过 DCT 变换后的一个系数矩阵。

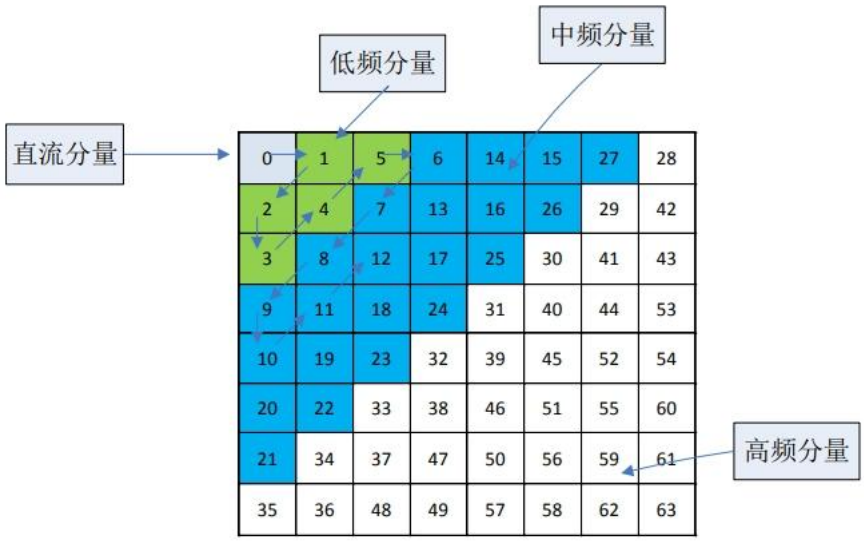
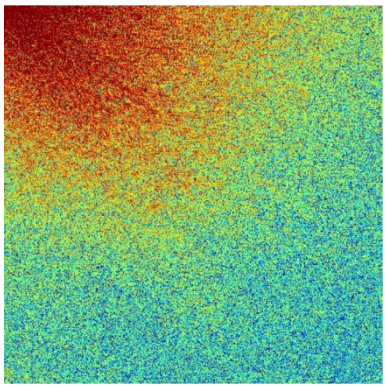


图 4.1 DCT 系数矩阵

使用 python 对 lema.bmp 图像做 DCT 变换，可以更加直观的看到 DCT 系数代表的能量分布，图 4.2 代表 lena 图像 DCT 变换后的能量分布图：

Lena 原始图像



DCT 变换后能量分布

图 4.2 DCT 变换

为了让图片在人类视觉中的尽量小，同时还要兼顾隐写效果的鲁棒性，一般选择在中频系数中隐藏数据。因为低频系数的改变会引起图篇画面的显著变化，高频会使隐写的鲁棒性会大大降低，因为压缩算法会破坏图片的高频信息，使隐写的数据丢失。所以要在隐蔽性和鲁棒性之间建立一个平衡，兼顾隐蔽性和鲁棒性，本文根据人类视觉系统的特点设计的水印算法就是为了平衡隐蔽性和鲁棒性。

4.2.3 DCT 水印嵌入方式

主要介绍三种的水印嵌入方式

（1）加法嵌入

如式(4.9)所示，加法嵌入中c_i是 DCT 系数矩阵中的第 i 个元素，现在要在这个元素中嵌入信息，就给元素上加上一个权重为 r 的的隐藏信息 w。

$$\mathrm{c}_i^{'} = \mathrm{c}_i + \mathrm{r}w \tag{4.9}$$

可以通过控制 r 的大小来控制嵌入强度，r 越大嵌入的强度越大，但是对原始图像的影响越明显。

（2）乘法嵌入

如式(4.10)所示，乘法嵌入中c_i是 DCT 系数矩阵中的第 i 个元素，现在要在这个元素中嵌入信息，就给元素上乘上一个一加权重为 r 的的隐藏信息 w。

$$\mathrm{c}_i^{'} = \mathrm{c}_i(1+\mathrm{r}w) \tag{4.10}$$

（3）对比嵌入

对比嵌入是选择 DCT 系数矩阵中的第 x 个元素和第 y 个元素，假设第 x 个元素的坐标是（a，b），第 y 个元素坐标是（c，d），现在要在这个图像分块中嵌入信息，比较 x 和 y 两个位置的元素的大小如果 $x>y$ 则为 1，如果 $y<x$ 则为 0。

三种嵌入方式各有利弊，加法嵌入增加一个固定的值，容易改变系数符号；乘法嵌入不会改变符号但是对矩阵中绝对值较大的系数影响过大；对比嵌入只改变系数矩阵中的两个系数，但是鲁棒性不强。平衡下本文选择加法嵌入的方式嵌入秘密信息。加法嵌入的强度系数将在 4.5.1 节中验证不同强度系数对侵入算法鲁棒性和不可见性的影响。

4.3 基于 HVS 的图像打分算法

前文介绍过，图像的 DCT 算法一般先将原始图像分成 8×8 的小块，秘密信息就隐藏到这些图像小块中。本节将介绍一种和 HVS 结合的分块图像的打分算法，目的是将所有的 8×8 小块按照是否适合隐写进行打分。从而实现分数越低的的（不适合隐写）排序越靠后，分数大的图像块（适合隐写）排序越靠前。通过排序来挑选隐藏秘密信息的图像块。

4.3.1 人类视觉系统 HVS

人类的视觉系统的特点在数字图像处理中有广泛的应用，这要是利用了人类视觉系统对图像频率、亮度、纹理、对比度有不同的感知特性。从而可以通过调整人类视觉系统不敏感的一些特性来隐藏信息并且几乎不会引起视觉系统的察觉。比如人类视觉系统

对纹理密集，亮度高的图像区域的细微变化不敏感，所以当嵌入载体图像的秘密信息低于人类视觉系统的对比门限(Contrast Sensitivity Threshold, CST)，眼睛就不会察觉到图像有修改的痕迹。本文主要利用了人类视觉系统的一些特性：

1.人类视觉系统对图像的边缘信息敏感，所以当图像区域有丰富的边缘信息，就要尽量少的去修改。

2.人类视觉系统对图像的平滑区域的变化十分敏感，所以在图像的平坦区域尽量少的嵌入信息。相反，如果图像区域有丰富的细节纹理，那么隐写数据带来的图像噪声和失真问题就会引起视觉系统的注意。

3.人类视觉系统对不同的灰度有不同的敏感性。人眼对中度的灰度最为敏感，然后对灰度高或者灰度低的部分不敏感，而且这种敏感度的下降是非线性的下降，所以要将秘密信息隐藏在图像的灰度高或者灰度低的区域。

基于上述的人类视觉来设计的嵌入方案，主要是通过分析图像的纹理，灰度，是否含有边缘信息来确定该图像区域的嵌入强度。从而自适应的嵌入隐藏信息。

4.3.2 分块图像排序方法

之前介绍了人类视觉系统的一些特性，比如果图像的纹理越复杂，背景的灰度偏高或者低，那么人类的视觉系统对图像的变化越不敏感，所以要嵌入秘密信息到图像中去就要嵌入到这种特征的图像块中去。可以通过分析图像的方差可以关联到图像的纹理，当方差大时，图像应该包含着比较复杂的纹理图案；分析图像的均值可以得到图像的整体亮度信息；可以通过 Sobel 边缘检测方式算子计算图像块的边缘纹理。下面将综合这三个条件对分块的图像进行排序，排序越靠前的越适合嵌入信息。

（1）计算分块图像的边缘纹理

图像的边缘检测方式有很多，这里选用了 Sobel 边缘检测，Sobel 边缘检测比 Candy 检测的准确性差但是效率却高。Sobel 边缘检测主要的原理是使用两个 3x3 的卷积核，分别检测每个像素水平方向梯度G_x和垂直方向的梯度G_y。通过将这两个方向的梯度幅值结合，可以得到边缘的整体强度。如式(4.11)所示，对所有分块图像进行式的量化，得到的边缘复杂度 E，E 越高的边缘越复杂。

$$E=\sum_{i=1}^{8}\sum_{j=1}^{8}(|G_x(i,j)|+|G_y(i,j)|) \tag{4.11}$$

计算出单个图像块的边缘复杂度后要进行归一化处理，这样才能确定该分块图像在所有图像分块中的复杂程度，也方便后续对每个图像块进行排序，归一化公式也很简单如式(4.12)所示，E_{score}表示归一化后的分数，*max(E)*表示所有分块中最大的边缘复杂度，

*min(E)*为最小的边缘复杂度，E_{block}表示当前图像块的边缘复杂度。

$$\mathrm{E}_{score}=\frac{E_{\mathrm{block}}-\min(E)}{\max(E)-\min(E)} \tag{4.12}$$

（2）计算分块图像的均值

HVS 对图像边缘纹理敏感，其次就是对亮度敏感。HVS 对于过亮或者过暗区域敏感度不高，针对这个特点，计算图片块的均值 μ。如式(4.13)所示，其中 M 和 N 都是 8，I_i和I_j表示的式图像中第 i 行第 j 列的灰度值。

$$\mu=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}I(i,j) \tag{4.13}$$

同样的给均值也要做归一化处理，如式(4.14)所示，μ_{score}表示归一化后的分数，μ_{block}表示当前图像块的均值。由于人眼对过暗或者过亮的区域不敏感，所以越靠近平均值越不适合隐写信息。

$$\mu_{score}=\frac{\left|\mu_{\mathrm{block}}-128\right|}{128} \tag{4.14}$$

（3）计算分块图像的方差

图片块的方差δ^2反映了图片的平滑程度，δ^2越大则图片块的包含这较多的纹理或边缘信息，这些地方适合隐写信息，当δ^2过小则反映了图块较为平滑，不适合隐藏信息。式(4.15)表示了图块的计算过程，其中 M 和 N 都是 8，I_i和I_j表示的式图像中第 i 行第 j 列的灰度值，μ表示图块的均值。

$$\sigma^2=\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(i,j)-\mu\right)^2 \tag{4.15}$$

同样对方差进行归一化处理。如式所示，$\delta^2{}_{score}$表示归一化后的分数，$max(\delta^2)$表示所有分块中最大的方差，$min(\delta^2)$为最小的方差，当最大和最小方差相同时$\delta^2{}_{score}$=0，$\delta^2{}_{block}$表示当前图像块的方差。

$$\sigma^2_{score}=\frac{\sigma^2{}_{\mathrm{block}}-\min(\sigma^2)}{\max(\sigma^2)-\min(\sigma^2)} \tag{4.16}$$

（4）计算分块图像的得分

上面介绍了给分块图片排序的三个重要指标，按重要程度分别是边缘纹理，亮度，纹理细节，并且已经对他们进行了归一化处理，现在要综合这三个评价指标，按照重要程度给每个参数设置权重，按照重要程度分别设置为α=0.5，β=0.3，γ=0.2。这样就得到了式(4.17)，数值越大则越适合嵌入图像。

$$\text{Score} = \alpha \cdot E_{score} + \beta \cdot \mu_{score} + \gamma \cdot \sigma_{core}^2 \tag{4.17}$$

根据式(4.17)，就可以计算所有分块图片的分数，将图片按照分数从小到达排列，我们就得到了一个分块图片序列 P。根据不同的分数设嵌入不同容量的秘密信息。下面章节会详细介绍。

4.4 图像水印的嵌入算法

前文已经介绍了结合 HVS 的图像块打分算法，通过这个算法我们可以量化每个图像块的隐写能力。下面将根据图像块的隐写能力，自适应的向载体图像中隐写数据。本节将从嵌入和提取一比特信息开始，介绍图像水印嵌入流程和其中的原理。

4.4.1 一比特信息的嵌入与提取

拿图像 lena 举例，lena 是 512×512 的灰度图像，将 lena 图像当作载体图像。对 lena 图像进行 8×8 的分块，那么 lena 图像一共被分成了 64×64 块，每一个小块隐藏 1 比特的秘密信息，那么 lena 图片一共可以隐藏 4096 比特的秘密信息。我们的秘密信息是 50×50 的二值图像，可以将秘密图像转成 2500 比特的 0，1 序列，白色像素为 0 黑色像素为 1。可以从 lena 图像中选取 2500 块来隐藏二维码序列的信息。 接下来将介绍如何在 8×8 的图像分块中隐藏 1 比特的信息，和如何将信息提取出来。

1.一比特信息的嵌入

嵌入过程如图 4.3 所示，首先判断嵌入的信息是 0 还是 1，当我们想在这个 8×8 的图像块嵌入 0 时，就不在这个 8×8 图像块嵌入任何信息；当我们想在这个图像块嵌入 1 时，我们就按照式(4.9)，将 22 个随机 0，1 序列（随机 0，1 序列的生成规则会在章节 4.4.2 介绍）按照一定的嵌入强度 r（r 的选取规则会 4.5.1 章节介绍）嵌入到图块的 22 个中频系数中，图 4.3 已经标注好了 DCT 变换后的中频系数位置。选接下来要介绍如何从图像块中提取刚刚嵌入的 1 比特信息。

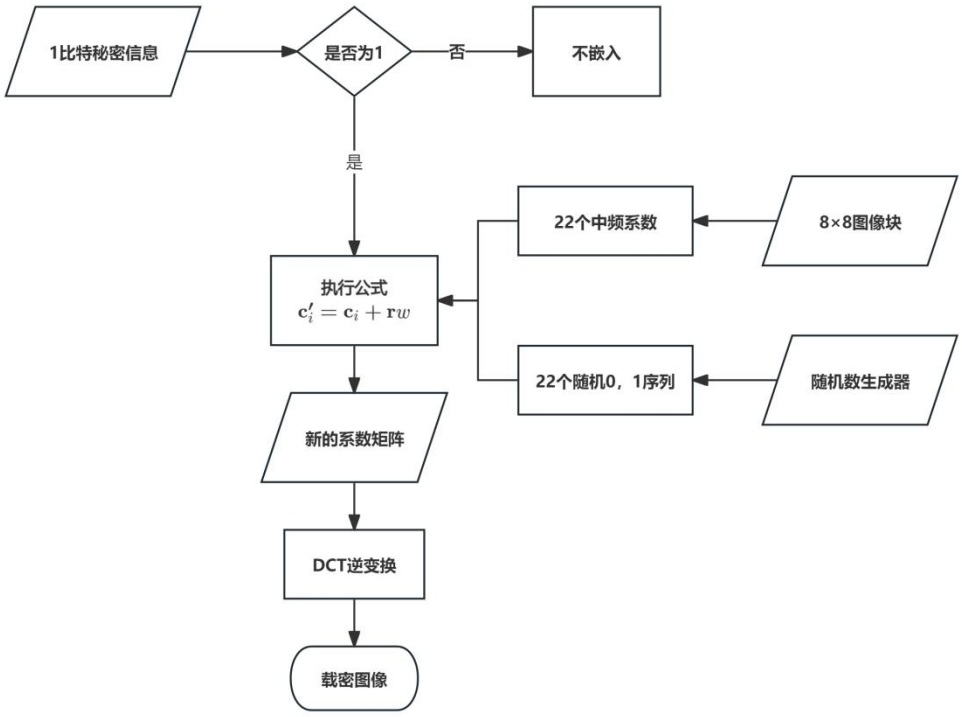


图 4.3 1 比特信息的嵌入

2.一比特信息的提取

信息的提取过程就是嵌入过程的逆过程。大体过程如图 4.4 所示，首先对原始图像和载密图像做 DCT 变换。选出要提取信息的 8×8 图像块，然后用载密图像块中频系数减去原始图像块中频系数，提取出嵌入向量。然后用归一化相关性来比较提取出的向量与原始向量的相似度，当相似度小于 0.5 的时候就代表没有嵌入水印，意味着该图像块代表比特 0；如果相似度大于了 0.5，则该像素块是嵌入了水印，图像块代表比特 1。相似程度我们用归一化相关系数（Normalized Correlation，NC），NC 的值越接近 1 则相似度越高。式(4.18)就是 NC 的数学表达式。

$$r = \frac{\sum (X - X)(Y - Y)}{\sqrt{\sum (X - \bar{X})^2} \cdot \sqrt{\sum (Y - \bar{Y})^2}} \tag{4.18}$$

其中 X, Y 为两个变量的观测值，$\bar{X}$和$\bar{Y}$为两变量的均值。

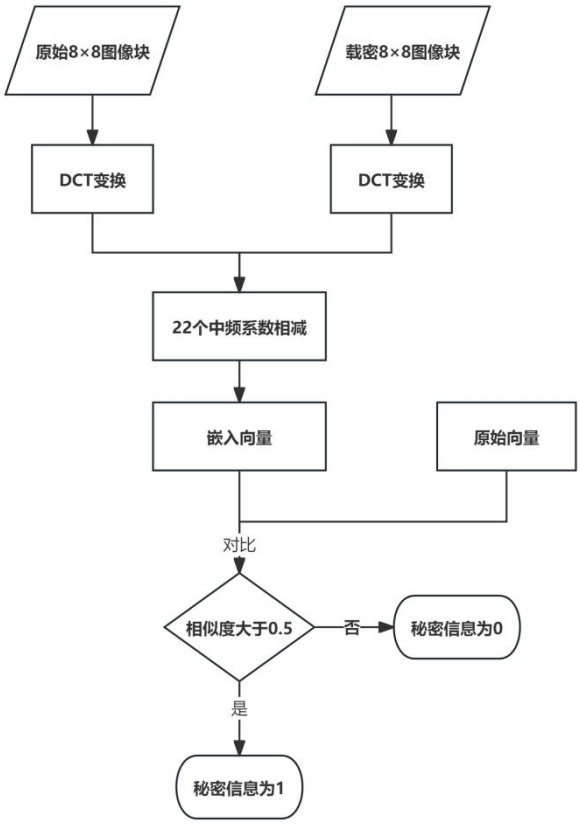


图 4.4　1 比特信息的读取

4.4.2　随机 0，1 序列的产生

在 4.4.1 中介绍了一个比特的秘密信息是如何嵌入和提取到 8×8 的图像块中的，在嵌入过程中需要用到一个随机 0，1 序列。对于我们要隐藏的秘密信息来说，一个 50×50 的二值图像，我们需要 2500 组随机的 22 位的 0，1 序列，对每一组随机序列的要求是：

（1）每组序列随机产生。

（2）序列长度为 22 位。

（3）数值为 1 的元素最少 8 位，最多 20 位。

以上的规定目的是分别是：

（1）保证序列的 0 和 1 的随机分布，这样尽量均匀的影响中频系数。

（2）因为选取的 22 位中频系数，所以对应 22 位的 0，1 序列。

（3）序列中为 1 的数可以改变中频系数，1 的数量越多对图像块的改变越大，可以为适合隐写信息的图像块分配 1 比较多的随机序列。

介绍了 0，1 序列的要求后，接下来介绍如何生成符合要求的 0，1 序列。在第三章我们详细介绍过一维混沌体系 Logistic-Tent 映射，也详细介绍了一位混沌序列的生成过

程。这里我们可以借助 3.3.2 节生成的一位混沌序列来产生我们需要的 2500 组 22 位的随机 0，1 序列，具体步骤如下。

步骤一：将 3.3.2 产生的一位混沌序列 S，前 10000 舍弃，保证序列的随机性

步骤二：从第 10001 位开始，向后选取 m×n×22 位元素，组成新的序列 S。m 是秘密二值图像的宽，n 是秘密二值图像的高。这里 m 和 n 都是 50。

步骤三：新的随机序列 S 每个元素都对 2 取模，将所有的元素都修改位 0 或 1。

步骤四：对 S 中的元素，从第一个开始，每 22 个分为一组。共分为 2500 组。

步骤五：对每一组统计 1 的个数，并按照 1 所含数量，按照从小到大的顺序排序。

步骤六：从第 1 组和第 2500 组，分别向数列中心统计 1 的数量，如果 1 大于 20 位，则将多余的 1 变成 0；如果 1 的数量小于 8 位，则将 0 变成 1，直到 1 的数量变为 8。

经过以上步骤就得到了最终的序列 S，S 包含了 2500 组按照的随机 0，1 序列，而且随着序列编号增多 1 的数量逐渐增多。

4.4.3 确定每一个比特的嵌入位置

在 4.2.1 中介绍了一个比特的秘密信息是如何在 8×8 的图像块中嵌入和提取的。现在要解决每个比特的秘密消息要嵌入到哪个图像块中的问题。对于要隐藏的秘密信息：一个 50×50 的二值图像。我们需要 2500 个图像块来嵌入秘密消息，通过 4.3.2 节，已经得到了排序过后的，包含 64×64 个图形块的序列 P，选择前 2500 个图像块嵌入消息。下面将介绍如何确定每个比特信息在哪个图像块中嵌入。

本节要达到的效果是：2500 个比特完全随机的散布在挑选出来的 2500 个图像块中，达到完全混乱的状态。这种需求和之前提到过的二维码的完全随机置乱几乎一样。两者都是要求，每个元素落在每个位置上的概率都是相同的。因此将采用同 3.4.2 节相同的方式来确定嵌入位置：

步骤 1：将秘密信息转换成成二进制序列 C。

步骤 2：利用 3.3.2 节生成的混沌序列 S 作为置乱序列。

步骤 3：从二进制序列 C 的最后一位像素开始循环

步骤 4：每次循环的像素位置为 i，每次循环 i 都减小一位。

步骤 5：利用混沌序列 S 生成随机下标 j，如式(4.19)所示。down()函数表示的是向下取整操作。

$$j=down(i*S(i)) \tag{4.19}$$

步骤 6：交换 J(i)和 J(j)

步骤 7：一直重复步骤 3 到步骤 6 到第一个元素，就得到置乱后的秘密信息。

4.4.4 水印的嵌入和提取步骤

前面小节我们已经介绍了如何将 1 比特信息在 8×8 图像分块中隐藏和提取。然后介绍了隐藏和提取过程中随机 0，1 序列的生成过程。下面将详细介绍如何将秘密二值图像隐藏到载体图像中，以及如何将秘密图像从载体图像中提取。秘密二值图像选择 50×50 的二维码图像，载体图像选择 512×512 的 lema 彩色图像。

1.水印的嵌入流程。

图像水印的嵌入流程如图 4.5 所示：

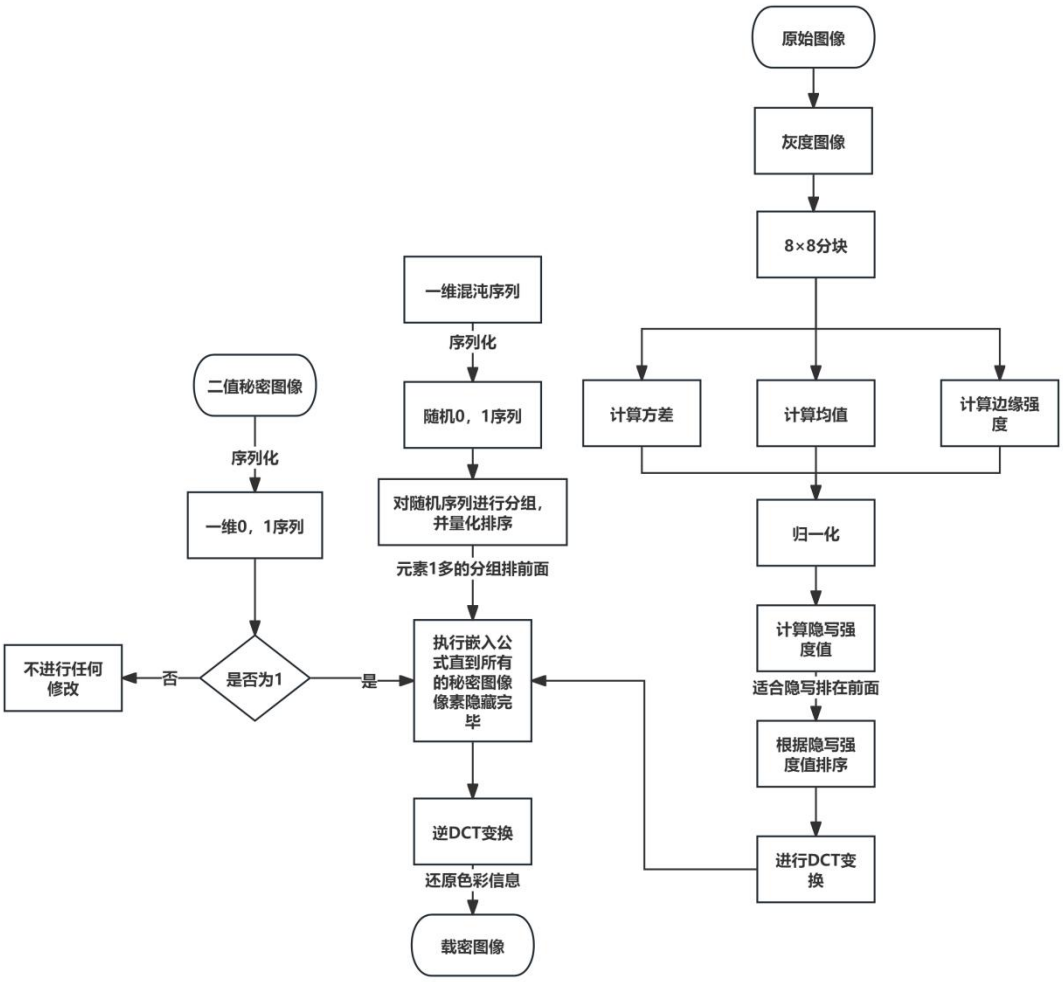


图 4.5 水印嵌入流程图

图 4.5 展示了秘密图像如何隐写到载体图像的全流程，下面将详细介绍所有步骤：

步骤一：对载体图像结合 HVS 进行排序。

（1）将载体图像按 8×8 分成互不重合的图像子块。

（2）计算每个图像子块的方差，均值，边缘强度，并且都做归一化处理。详细步

骤已经在 4.3.2 详细介绍。

（3）对每个子块计算适合隐写的分数，按分数排序，分数越高越适合隐写，排序越靠前。得到排序后的图块序列 *P*。

步骤二：将二值秘密图像进行序列化。

（1）将二值图像转成一维序列，白色像素块代表 0，黑色像素块代表 1。

（2）得到二值图像的一位序列 *C*。

步骤三：将混沌序列映射成随机 0，1 序列：将 3.2.2 计算出的混沌序列 *S* 进行映射，生成符合要求的新的随机 0，1 序列。具体步骤已经在 4.4.2 中详细给出。

步骤四：对步骤一已经排序好的图块序列 *P* 所有的图块做 DCT 变换。

步骤五：查看 *C(i)*是否为 0，进入步骤六；如果 *C(i)*为 1，则进入步骤七。

步骤六：*P(i)*无需做任何操作，查看 *i* 是否为 *C* 序列的最后一个元素。若不是，则 *i* 增加一位，重复步骤五；若是最后一位，执行步骤八。

步骤七：对图块 *I(i)*执行 1 比特的嵌入操作，嵌入操作在 4.4.1 已经详细介绍。并查看 *i* 是否为 *C* 的最后一个元素。若不是则 i 增加一位，重复步骤五；若是最后一位，执行步骤八。

步骤八：对图片序列 *P* 所有的图像块做逆 DCT 操作，并将所有的图像块放回原始位置，组合为图像 *E*。

步骤九：将图像 *E* 的色彩通道重新融合，得到载密图像。

2.水印的提取流程。

图 4.6 表示了提取水印的所有流程，如图所示：

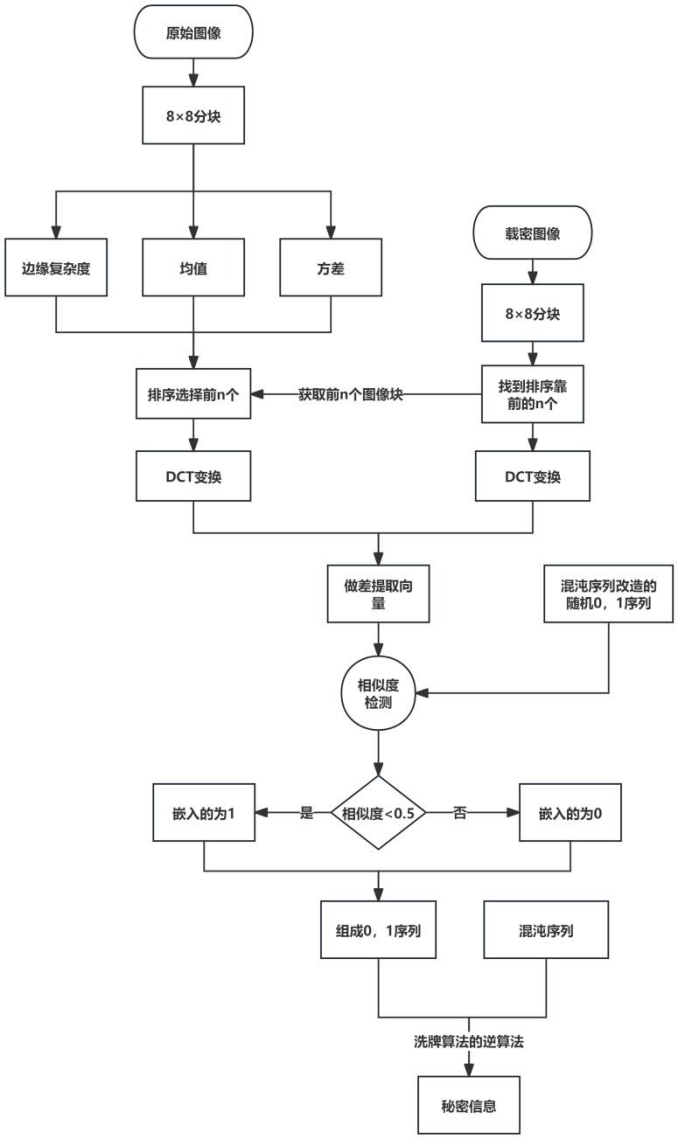


图 4.6 水印提取流程图

图 4.6 展示了秘密图像如何从载体图像提取的全流程，下面将详细介绍所有步骤：

步骤一：对原始图像结合 HVS 进行排序。

（1）将原始图像按 8×8 分成互不重合的图像子块。

（2）计算每个图像子块的方差，均值，边缘强度，并且都做归一化处理。详细步骤已经在 4.3.2 详细介绍。

（3）对每个子块计算适合隐写的分数，按分数排序，分数越高越适合隐写，排序越靠前。得到排序后的前 2500 个图块序，这些图像块组成图像块序列 P。

步骤二：找到载密图像中作为嵌入信息的图像块。

（1）序列 P 中的图像块位置就是载密图像中嵌入信息的图像块的位置。

（2）将载密图像中 2500 个载密图像块按照 P 序列排序

步骤三：载密图像块和原始图像做差并除以嵌入嵌入 r 得到了提取向量$\vec{a}$，将向量$\vec{a}$和对应的随机 0，1 序列中实际嵌入向量$\vec{b}$做相似性计算，如果相似性小于 0.5 则当作没有嵌入信息，代表比特 0；如果相似性大于 0.5 则嵌入了信息，代表比特 1。

步骤四：一直循环步骤三，直到 2500 个载密的像素块的嵌入的 0，1 比特都被提取出来。

步骤五：按照混沌序列将 2500 个 0，1 序列做洗牌算法的逆算法，恢复实际的序列位置。

步骤六：将步骤五恢复的一位序列恢复成二维的二值图像，秘密信息提取完成

4.5 实验结果和分析

本章选择的载体图像是 512×512 的灰度图像 Lena，秘密图像选择的是 48×48 的二维码图片，二维码图片选择最高的纠错等级（30%），如图 4.7 所示，实验是将图片 b 嵌入到图片 a 中，然后对载密图片 a 进行攻击操作，最后提取并解密为图片 c。

首先确定在没有攻击的情况下，不同的嵌入强度系数对图像的影响。随后对载密图像继续裁剪攻击、椒盐噪声攻击、滤波攻击、直方图攻击、JEPG 压缩攻击。来测试嵌入算法的鲁棒性。解密后的二维码是否可读可以直接通过微信扫一扫查看是否扫出“hello”的 SHA256 算出的哈希值。

a　　b　　c

图 4.7 （a）Lena 图像的原始图，（b）加密后的数字水印，（c）数字水印

4.5.1 不同嵌入强度的嵌入实验

在按照前面的嵌入算法，选择嵌入强度分别为 1、5、10、15、20、25、30、35、40、45、50 将二维码嵌入到载体图像中，接下来做两个对比实验：

第一个对比实验：比较载密图像和载体图像得到他们的峰值信噪比（PSNR），公

式见式(3.26)。当 PSNR 在 25dB 与 50dB 之间的时候，HVS 几乎不能感觉出图像明显的不同，通过对比 PSNR 我们可以知道当何种嵌入强度的时候图像会有明显的失真。

第二个对比实验：提取不同嵌入强度下的水印，得到不同嵌入强度下提取的水印图像。然后对比原始水印图像和提取后水印图像的归一化相关系数（Normalized Correlation，NC）。NC 是用来比较两个图片的相似程度，公式见式(4.18)。

经过实验，得到的不同嵌入强度系数的情况下，载密图像的 PSNR 和水印图像的 NC 值如下表 4.1 所示：

表 4.1 嵌入强度与 PSNR 和 NC 的关系对比

嵌入强度	1	5	10	15	20	25	30	35	50
PSNR	60.254	50.248	42.642	39.568	37.256	31.245	28.124	22.365	12.368
NC	0.9884	0.9886	0.9945	0.9945	0.9945	0.9945	0.9945	0.9945	0.9945

从表中可以分析得出：嵌入强度越大，水印对图像的影响也就越大，PSNR 值越小。NC 值在不受任何攻击的情况即使嵌入强度很小也能很好的提取出完整的信息。在没有任何攻击的情况下嵌入强度可以选择很小的值，对原始图像的干扰也最小。

4.5.2 裁剪攻击

剪切攻击我们选择剪切坐上角剪切 1/4，图 4.8 展示的是嵌入强度为 35 的情况下，剪切后的载密图像以及，提取完加密数字水印（像素太小不易扫码，已经将图像 b 等比例放大，后面展示的二维码均以放大），并将数字水印解密后的状况。

a b

图 4.8 剪切 1/4 后提取并解密水印

从表 4.2 可以看出，剪裁 1/4 后仍能还原数字水印，并且可以扫码还原出二维码内

容。

表 4.2剪切 1/4 在不同嵌入强度的 NC

嵌入强度	1	5	10	15	20	25	30	35	50
NC	0.2484	0.3436	0.4345	0.4535	0.4941	0.5465	0.5875	0.6141	0.6345

如表 4.2 所示：随着嵌入强度越大，二维码的提取质量越高。嵌入强度低于 30 的已经无法读出二维码的内容。

4.5.3 椒盐噪声攻击

在图片传输的过程中很容易出现信道不稳定从而导致噪声干扰的情况。用椒盐噪声攻击来模拟图片在噪声下的情况，这里是给载体图片添加了 0.02 的椒盐噪声，水印的嵌入强度是 35。

a

b

图 4.9 载体图像添加 0.05 的椒盐噪声

从图 4.9 可以看出，在嵌入强度为 35 的情况下二维码内容可以读出来。

表 4.3 0.02 的椒盐噪声攻击下的 NC 值

嵌入强度	1	5	10	15	20	25	30	35	50
NC	0.2081	0.3036	0.4145	0.5335	0.6041	0.6441	0.7107	0.7502	0.8245

如图 4.9 所示：嵌入强度低于 15 的已经无法读出二维码的内容。

4.5.4 JPEG 压缩攻击

因为我们修改的是中频系数，jpeg 压缩的原理是对高频系数的压缩，所以理论上 DCT 的图像嵌入算法具有良好的抵御 jpeg 压缩的能力。我们准备不同的压缩因子，分

别为 80%、60%、40%、20%，来测试算法抗 jpeg 攻击的能力。这次直接选取 35 的嵌入强度。

a b c d

图 4.10 不同压缩因子提取数字水印

根据图 4.10 展示，a，b，c，d 分别是压缩因子 80%、60%、40%、20%。在嵌入强度 35 的条件下提取出的二维码。证明 40%的压缩因子下仍然可以读出嵌入信息。20%不可读。

4.6 本章小节

本章首先介绍了离散余弦变换（DCT）的基本概念，对 DCT 变换在水印嵌入方面的原理进行了介绍。随后提出了一种结合人类视觉系统（HVS）的图像打分公式：基于人类视觉系统对图像不同特性的敏感程度不同（边缘复杂度>亮度>纹理复杂度），将这三个变量都进行归一化处理，选取不同的权重，最后得到一个归一化分数。通过这个分数我们可以将载体图像的每个 8×8 分块都量化评价是否适合嵌入秘密信息，从而对所有的图像分块排序，排序越靠前优先嵌入秘密信息而且信息的嵌入量越大。

随后本章开始介绍具体的二值图像嵌入算法，先从一个比特的秘密信息嵌入和提取讲起：将载体图像进行 8×8 分块，每一个 8×8 的图像块嵌入 1 比特的信息。嵌入方式是将图像块的 22 个中频系数和 22 个随机 0，1 序列进行加法嵌入，嵌入 1 的时候进行加法嵌入，嵌入 0 的时候不做操作。随后介绍了 0，1 序列的生成规则：通过 logistic-tent 混沌序列和用户密钥生成的混沌序列，将混沌序列归一化之后按 22 比特分组，并将每个分组中 1 的数量控制在一个范围内，随后按照 1 的数量将分组排序，这样做的目的是让打分高的载体图像块能嵌入更多的信息。最后一步是将二值水印序列化，并通过洗牌算法和混沌序列确定每个比特嵌入到哪一块图像块中。当每个比特都嵌入完毕后进行逆 DCT 变换，添加图像的 RGB 通道，完成水印嵌入。

随后通过实验确定了确定在没有攻击的情况下的不同的强度系数 r 和算法不可见性。然后对嵌入图像添加裁剪攻击，校验攻击，JPEG 压缩攻击，测试不同嵌入强度下的算法的鲁棒性。

第五章　总结与展望

5.1 工作总结

随着信息技术的发展，人们在网络上的产生大量的图片信息需要存证，常用的做法是将图片的哈希值和元数据信息发送到一个第三方可信存证机构。本文在此方案的基础上增加了一个本地化的数字水印，利用用户上传的哈希值和元数据在生成一个唯一校验码嵌入到原始图片中。这样即使用户图片被修改盗用也可以通过本地的水印来确认是否侵权。，就需要对水印进行加密操作，此外嵌入算法需要兼顾鲁棒性和不可见性：当载密图片受到裁剪、污损，压缩后仍能提取水印。而且没有用户的密钥难以将水印提取成功。

为了解决水印信息被窃取的问题，本文设计了二值水印的加密模型，可以将带有版权信息的二值水印图片完全置乱。为了解决嵌入算法的鲁棒性和不可见性，以及需要密钥才能提取水印的需求，本文设计了基于 HVS 和混沌系统的 DCT 域二值水印嵌入算法。具体工作如下所示。

（1）设计了结合混沌系统和二维码的数字水印的生成和加密方案：该方案将 QR 码和混沌系统结合，将用户上传的图片哈希值和元数据信息利用 SHA256 算法生成唯一的散列值 S。并将散列值 S 生成为 QR 码。然后结合 logistic-tent 系统和洗牌算法将 QR 码完全置乱，隐藏 QR 码的像素的位置信息。随后利用 Chen 混沌系统和 DNA 编码扩散算法将置乱后的 QR 码进行扩散操作，消除 QR 码的黑白像素数量信息。经过以上操作得到的完全随机无序，加密的二值数字水印。最后对数字水印的生成和加密方案进行实验，论证本方案的算法可逆性，密钥敏感性，在裁剪、椒盐噪声等攻击下的加密鲁棒性。

（2）结合了人类视觉系统（HVS）和混沌系统的 DCT 域二值图像嵌入算法。具体做法是：将原始图片进行分块，依据人类视觉系统对图片边缘复杂度、纹理复杂度、亮度的不同敏感性，给每块图像打分并排序，分数越高的图像块越适合做水印的嵌入。随后将二值数字水印序列化成 0，1 比特序列，并结合用户提供的密钥和 logistic-tent 混沌系统随机确认每个比特嵌入哪个图片块。在嵌入过程中基于改进的 DCT 域的数字隐藏算法，越适合嵌入信息的图像块嵌入系数越大，修改幅度越大。以此来平衡数字水印的鲁棒性和不可感知性。最后进行实验，验证算法可逆性，分析在不同嵌入强度下对鲁棒性和不可感知性的影响。

5.2 未来工作展望

尽管本文完成了一部分工作，但是还存着一些不足需要完善。有以下的内容可以在后续的工作中提升优化。

（1）随着图片信息的暴增，单纯依靠 256 位的唯一校验码难满足标记每一张图片且不产生碰撞。所以可以设计一种认证方式，在巨大存证容量的情况下不会出现碰撞。

（2）对图像的处理可以利用 GPU，加快对图片水印的加密和嵌入操作，提高效率。

参考文献

[1] Lorenz E. Deterministic nonperiodic flows[J]. Journal of the Atmospheric Sciences, 1963,20:267-285.

[2] Ozturk I, Sogukpinar I. Analysis and comparison of image encryption algorithms[J]. International Journal of Information Technology, 2004, 1(2): 108-111.

[3] Yan S, Gu B, Wang E, et al. Finite-time synchronization of multi-scroll hyperchaotic system and its application in image encryption[J]. Mathematics and Computers in Simulation, 2023, 206: 391-409.

[4] Zhou Y, Bao L, Chen C. A new 1D chaotic system for image encryption [J]. Signal Processing, 2014,97(7):

[5] Hua Z, Jin F, Xu B, et al. 2D Logistic-Sine-coupling map for image encryption[J]. Signal Processing, 2018, 149: 148-161.

[6] Lorenz Edward N. Deterministic Nonperiodic Flow[J]. Journal of the Atmospheric Sciences, 1963, 20(2): 130-141.

[7] Kuate P D K, Lai Q, Fotsin H. Complex behaviors in a new 4D memristive hyperchaotic system without equilibrium and its microcontroller-based implementation[J]. European Physical Journal Special Topics, 2019, 228(10): 2171-2184

[8] H. B, Z. H, N. W, et al. Initials-Boosted Coexisting Chaos in a 2-D Sine Map and Its Hardware Implementation[J]. IEEE Transactions on Industrial Informatics, 2021, 17(2): 1132-1140.

[9] AlShaikh M,Alzaqebah M,Gmati N, etal. Image encryption algorithm based on factorial decomposition[J]. Multimedia Tools and Applications, 2024, DOI10.1007/s11042-023-17663-1.

[10] Wen H, Lin Y. Cryptanalysis of an image encryption algorithm using quantum chaotic map and DNA coding[J]. Expert Systems with Applications, 2024, 237: 121514.

[11] Huang C. et al., "Research on the Development of Chaotic Image Encryption", 顶刊综述, 2023.

[12] Mixed Multi-Chaos Quantum Image Encryption Scheme Based on QCA, IEEE Trans. Quantum Eng., 2023.

[13] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps[J]. International Journal of Bifurcation and chaos, 1998, 8(06): 1259-1284.

[14] 陈锦彬，叶瑞松,基于改进 Henon 映射的混沌图像加密算法[计算机科学与应用,2022,12(2):422-435.

[15] Li T., "Hyperchaotic Image Encryption Based on Multiple Bit Permutation and Diffusion", Elsevier J. Inform. Security, 2022.

[16] Qian K. et al., "忆阻混沌系统与双向比特循环移位方案", Frontiers in Physics, 2022.

[17] Wang X. et al., "2D Sine-Logistic-Tent-Coupling Map for Image Encryption", IEEE Access, 2023.

[18] Zhang J , Lu Z , Li M .Study on an efficient hyper-chaos-based image encryption scheme using global

bit permutation[J].IOS Press Open Library, 2020, 28(Suppl 1).DOI:10.3233/THC-209030.

[19] Neil F. Johnson, Sushil Jajodia. Exploring Steganography: Seeing the Unseen.Computer,1998, 31(2): 26-34

[20] Benedikt Boehm.StegExpose-A Tool for Detecting LSB Steganography.arXiv,2014 ,1410.6656

[21] Bin Li, Ming Wang, Jiwu Huang, Xiaolong Li. A new cost function for spatial imagesteganography.In: 2014 IEEE International Conference on Image Processing (ICIP2014). IEEE, Paris, France, October 27 -30, 2014: 4206-4210

[22] Vojtech Holub ,Jessica J. Fridrich. Designing steganographic distortion usingdirectional filters.In: 2012 IEEE International workshop on information forensics andsecurity (WIFS 2012). IEEE, Costa, Adeje, December 2-5, 2012: 234-239

[23] Denemark T, Fridrich J, Holub V. Further study on the security of S-UNIWARD.Media Watermarking, Security, and Forensics, 2014, 9028: 902805

[24] Weng CY.DWT-based reversible information hidingscheme usingprediction-error-expansion in multimedia images[J]. Peer-to-Peer Networking and Applications,2020,13(2):514-523.

[25] 鲁业频，李凤亭，陈兆龙，朱仁义. 离散余弦变换编码的现状与发展研究. 通信学报，2004(02)：106-118

[26] 桑军,向宏,胡海波.基于离散傅里叶变换的信息隐写术.华中科技大学学报:自然科学版,2008,36(8):5-8.

[27] Quan L X. A Research on Information Hiding Algorithm Based on FrequencyBlocks of DCT Coefficients[P].Management Engineering,2019,72:47-50.

[28] Kundur, D Hatzinakos.A robust digital image watermarking method using wavelet-based fusion. In: Proceedings of the lEEE International Conference on ImageProcessing, ICIP 97,1997:544-547

[29] Ariatmanto D, Ernawan F. Adaptive scaling factors based on the impact of selected DCT coefficientsfor image watermarking[J]. Journal of King Saud University-Computer and Information Sciences,2022, 34(3): 605-614

[30] Liu Y, Tang S, Liu R, et al. Secure and robust digital image watermarking scheme using logistic andRSA encryption[J]. Expert Systems with Applications, 2018, 97:

[31] 马婷,李佳.基于混沌加密的 QR 码彩色图像复合水印算法[J].印刷与数字媒体技术研究,2023,(02):50-56+82.

[32] 周希.基于量子离散余弦变换的彩色图像水印算法[D].南昌大学,2023.

[33] 刘伟宏,谢维信,喻建平等.基于人类视觉系统的自适应数字水印算法.上海交通大学学报.2008.42(7):1144-1148.

[34] 沈磊,周鹏颖,田小林等.基于人类视觉系统和 DCT 的数字水印算法.微计算机信息.2010.26(6-2):212-213.

[35] 傅俊.QR 码图像信息隐藏技术及其隐秘通信系统研究与实现[D].湖南大学,2017

[36] 赵博,黄进.基于 PDF417 条码的信息隐藏方法[J].计算机工程与设计,2007,28(19): 4806- 4809.

[37] 吴佳鹏. 二维条码识读技术及其应用研究[D]. 博士论文, 天津大学, 2009

[38] ISO/IEC 18004:2006.Information technology-Automatic identification and data capture technology QR code 2005 bar code symbology specification[S].

[39] Sartid Vongpradhip, Suppat Rungraungsilp. QR Code Using Invisible Watermarking in Frequency Domain[C]. 2011 Ninth International Conference on ICT and Knowledge Engineering.pp.47-52,2011.

[40] Gleick J. Chaos: making a new science[M].Viking: 1987.

[41] Li T Y, Yorke JA. Period three implies chaos[M]. Springer, 2004.

[42] Young L S. Dimension, entropy and Lyapunov exponents[J]. Ergodic Theory and Dynamical Systems, 1982, 2(1): 109-124

[43] Wolf A, Swift J B, Swinney H L. Determining Lyapunov exponents from a time series[J]. Physica D: nonlinear phenomena, 1985, 16(3): 285-317.

[44] Lorenz E. Deterministic nonperiodic flows[J]. Journal of the Atmospheric Sciences,1963,20: 267-285.

[45] Shannon C E. Communication theory of secrecy systems[J]. The Bell system technical journal, 1949, 28(4): 656-715.

[46] Mohammed Basna, Ameen Siddeeq, Omer Hassan. Image authentication based on watermarking approach: Review[J]. Asian Journal of Computer Science and Information Technology, 2021, 9(3): 34-51.

[47] Swaraja K, Meenakshi K, Kora Padmavathi. Hierarchical multilevel framework using RDWT-QR optimized watermarking in telemedicine[J]. Biomedical Signal Processing and Control, 2021, 68(7): 102688-102701.

[48] 秦如贞,张黎明,伍庭晨,等.结合 ASIFT 和归一化的抗仿射变换遥感影像盲水印算法[J].地球信息科学学报, 2021,23(10):1882-1891.

[49] Gaurav Verma, Meihua Liao, Dajiang Lu, et al A novel optical two-factor face authentication scheme[J]. Optics and Lasers in Engineering, 2019, 12(123): 28-36.