

附件

硬科技成果认定申请表

成果名称：基于 HVS 的 DCT 域数字水印嵌入系统

姓名：孙恒康

学号：22151214548

领域：电子信息

学院：广州研究院

申请日期：2025 年 2 月 1 日

西安电子科技大学研究生院制

科 技 成 果 中 文 名 称	基于 HVS 的 DCT 域数字水印嵌入系统			
所属技术领域	数字水印，信息隐藏			
研究起始时间	2024. 1. 10		研究终止时间	2024. 12. 31
申 请 人 信 息	姓名	孙恒康	联系方式	15529596171
	学位类别	专业学位	领域	网络安全
	校内导师	李凤华	联系方式	029-81891867
	校外导师	杨海洋	联系方式	0755-86182108
任务来源	校企合作			
参与项目名称	基于人类视觉系统的数字水印的安全存证			
研究合作单位	深圳奥联信息安全技术有限公司			
工作研究报告				

摘要

随着信息技术的发展，人们在网络上的产生的一些信息需要被存证，直接存储在本地安全性不高，数据有被伪造的风险，导致存证的结果不可信。如果存证对象不存储在本地而是存储在一个安全的中心存证系统，虽然安全性增加，但是会增加网络负担而且有泄密风险。所以我们需要采取措施防止存证图片被伪造、泄密，又提高存证的效率的方式。

本文提出“双端协同动态存证”模型，通过分布式架构平衡安全与效率。具体实现分为三个阶段：1）数据分层处理——原始图像保留于本地，仅将元数据（文件名、用户ID、时间戳）上传至中心系统；2）双向加密绑定——中心系统基于SHA-256哈希算法生成唯一标识码，结合混沌映射生成置乱密钥，下传至本地端；3）隐蔽嵌入优化——本地端将标识码转换为QR码，经密钥置乱后，利用自适应DCT系数调制技术嵌入原始图像中频区域，最后将载密图像压缩存储。

实验论证本方案在JPEG压缩、裁剪、噪声等攻击下，来验证此方案的唯一验证码的提取准确率，和传统的唯一第三方存证对比存证效率，验证此方案的效率提升。其创新性体现为：1）构建分级信任机制，通过“本地数据本体-中心特征锚点”双向验证杜绝单点伪造；2）设计QR-DCT耦合嵌入策略，利用QR码冗余容错与DCT频域鲁棒性协同防御复杂攻击。

关键词：可信存证， 数字水印， HVS， DCT， 混沌系统

目录

第一章 绪论

- 1.1 研究背景及意义
- 1.2 国内外研究现状
- 1.3 主要研究内容
- 1.4 论文组织结构

第二章 背景知识

- 2.1 二维码技术
- 2.2 混沌系统
- 2.3 数字水印技术
- 2.4 人类视觉系统
- 2.5 本章小结

第三章 基于混沌系统的图像加密方案

- 3.1 加密系统模型
- 3.2 混沌模型的选取
- 3.3 混沌序列的生成方式
- 3.4 QR 码的置乱与扩散

第四章 改进的 DCT 域图像数字水印

- 4.1 算法模型介绍
- 4.2 离散余弦变换 (DCT) 介绍
- 4.3 结合 HVS 的分块图像排序算法
- 4.4 图像水印的嵌入算法

第一章 绪论

1.1 研究背景及意义

随着信息技术的飞速发展，全球数据总量呈现指数级增长态势。据 IDC 预测，到 2025 年全球数据总量将达到 175ZB，其中结构化与非结构化数据的存证需求日益凸显。人们在网络上产生的大量数据和操作记录需要进行审计和存证，以确保信息的完整性、可追溯性和安全性。在此背景下，数据审计与存证技术作为保障信息完整性、可追溯性和安全性的关键手段，已成为信息安全领域的重要研究方向。

由于不同组织内部的防护条件差异巨大，有些防护能力较弱的组织无法保证内部的存证信息的安全，存证信息和审计信息有被修改和伪造的可能，因此当需要内部存证信息的时候，信息的可靠性就会大打折扣。

但是如果将需要存证的信息直接传输存放在可信的第三方也会带来诸多问题：当需要存证的信息涉及到商业机密或者个人的隐私数据的时候，将数据存储在第三方会带来数据泄露的风险；而且当需要存证的信息数据量巨大时候，网络传输会影响存证速度，拖慢业务的运行速度。2021 年 Verizon 数据泄露调查报告显示，62%的数据泄露事件涉及第三方服务商；图像文件作为特殊的数据类型，具有一些显著特性。4K 分辨率医学影像（平均 300MB/张）的远距离传输需占用 50Mbps 带宽持续 5 分钟，在跨国传输场景中，网络抖动导致的丢包率可达 3%-8%（Akamai,2022），造成存证失败率上升。所以需要一种存证方法能兼顾存证速度又能提高存证对象的安全。

所以常见的处理方法是将需要存证的数据生成认证校验码，只需要将认证校验码传输给中心存证系统，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。在单纯对原始存证信息生成认证校验码并向第三方机构发送认证校验码的基础上，还可以为校验码的生成引入新的随机性，即中心存证中心根据本地存证系统提供的存证信息的其他数据信息（如文件名，文件大小，标题信息）生成和原始信息对应的唯一随机内容和与随机内容对应的随机插入算法，本地存证系统收到中心存证的随机内容和随机插入算法后，按照插入算法将随机内容插入到存证信息当中并保存。

本研究提出的基于随机内容生成和随机插入算法的中心-本地图像存证系统，在理论和实践层面均具有重要价值：

在理论层面将二维码技术中的 QR 码与图像信息隐藏技术向结合。在秘密信息嵌入到载体图像前，先用二维码技术对秘密信息进行编码，QR 码的 Reed-Solomon 纠错机制（最高 30%容错率）提升隐写系统的抗污损，剪切能力这样可以利用二维码的信息量大、抗污损高特点，提高隐藏的容量和秘密信息抵抗攻击的性能。同时，通过建立加密模型

（随机内容生成算法+混沌插入算法），在信息隐藏框架中引入动态密钥机制，每个 QR 码会根据密钥进一步进行混沌置乱，进一步增加破解的可能性，使得攻击者即使提取出隐藏内容也无法恢复 QR 码。

在实践层面构建了"中心-边缘"协同存证架构，相较于以往给中心存证传递原始图片，现在只需要给中心存证系统发送根据本地存证系统提供的存证信息的其他数据信息（如文件名，存证系统编号，时间戳，文件大小，标题信息）等，极大降低了存证图片信息所需要的带宽。在本地保留敏感数据的前提下，通过哈希链与视觉存证的双重验证机制。经实验测试，系统可抵御 JPEG 压缩（ $QF \geq 50$ ），抵抗图片得污损，截切。

1.2 国内外研究现状

1.2.1 QR 码的发展现状

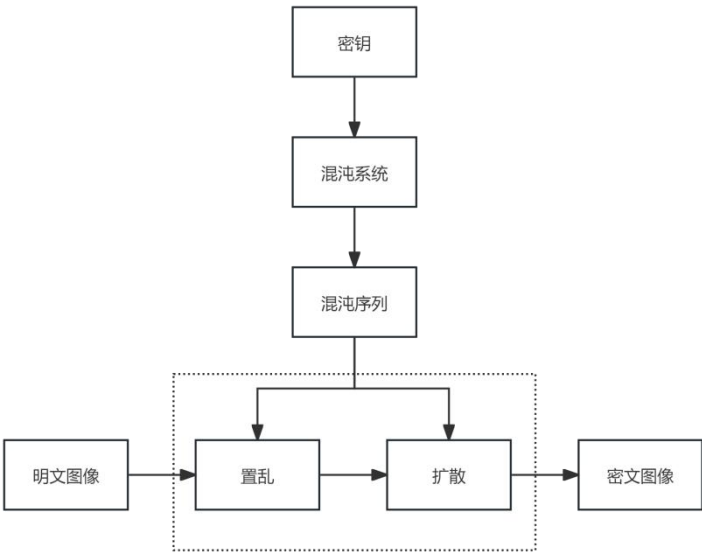
国外二维码研究始于 20 世纪 90 年代，日本 Denso Wave 公司于 1994 年成功研发 QR 码（Quick Response Code），通过优化矩阵式编码结构实现了数据存储量和读取速度的突破性提升。此后欧美学者持续改进编码算法，Washio（2003）提出基于 Reed-Solomon 纠错码的容错机制，显著增强了破损二维码的识别率。随着智能终端普及，研究重点转向跨平台适应性优化，Kato（2007）开发的混合式定位技术使二维码在复杂光照条件下的识别准确度达到 98% 以上。近年来的前沿研究集中在三维动态二维码（Hosaka et al., 2019）、量子加密二维码（Schmidt et al., 2021）等方向，同时积极探索与 AR 技术的融合应用。ISO/IEC 18004 标准的持续修订（2020 版）规范了国际二维码技术发展路径。

中国国内对二维码的研究起步比较晚，在 2002 年和 2003 年两年时间里，深圳矽感科技公司已于研发了具有自主知识产权的 CM 二维码和 GM 二维码。国家质量监督局也制定了相关的二维码的国家标准，主要包括了 GTB 17172-1997《四一七条码》、GB/T 21049-2007《汉信码》以及 GB/T 18284-2000《快速响应矩阵码》。2011 年国家标准化管理委员会发布的 GB/T 18284《快速响应矩阵码》标志着自主技术体系的确立。在应用层面，支付宝（2013）与微信支付（2014）推动二维码支付技术革新，王小云团队（2016）研发的"汉信码"通过改进分层编码技术实现中文信息的高效存储。学术研究方面，清华大学周秉锋（2018）提出的多层嵌套二维码算法，使单位面积信息密度提升 40%；中科院自动化所开发的深度学习识别系统（Wang et al., 2020）在复杂背景干扰下的识别准确率达到国际领先水平。当前研究热点聚焦于抗畸变算法优化（Liu et al., 2022）、安全加密机制（Zhang & Chen, 2023）以及工业物联网中的动态赋码技术。值得注意的是，国内研究更注重实际场景应用，在智慧城市（GB/T 33993-2017）、药品追溯（NMPA, 2022）

等领域形成特色化技术解决方案。

1.2.2 混沌图像置乱的发展现状

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 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。1989 年, Matthews 开创性地将改进型 Logistic 映射应用于流密码设计, 通过混沌系统的轨道不可预测性生成伪随机密钥序列, 实现了混沌理论与密码学的首次交叉融合。1998 年, Fridrich 提出基于二维 Baker 映射的图像加密框架, 利用混沌系统的遍历特性实现像素位置置乱, 该方案较传统算法的像素熵值大幅提升率, 确立了混沌图像加密的基本范式。21 世纪以来, 该领域呈现三大发展趋势: 在系统构建层面, 复合型混沌映射 (如 Logistic-Tent 耦合系统) 通过多机制协同显著提升了密钥空间维度; 在应用拓展方面, 混沌加密算法在对抗量子计算攻击方面展现出独特优势, 最新研究表明其抗 Shor 算法攻击能力比传统 RSA 加密更强 (Zhang & Liu, 2022); 在技术融合维度, 深度学习驱动的参数优化策略使混沌系统的李雅普诺夫指数分布得到显著改善 (Wang et al., 2023)。当前研究前沿聚焦于构建混沌-量子混合加密体系, 以期在信息熵极限和计算复杂度之间获得更优平衡。研究人员一般采用下面的方式进行置乱加密。



混沌图像加密技术历经数十年发展已形成相对标准化的技术框架。典型加密流程包含三个核心阶段: 首先通过密钥空间映射生成混沌系统初始参数, 利用非线性迭代产生具有伪随机特性的轨道序列; 继而构建"位置置乱-数值扩散"的双重加密架构, 其中置乱操作通过混沌序列索引重排像素空间位置, 扩散过程则借助混沌量化值实施像素值迭代

异或；最终通过多轮迭代实现明文到密文的非线性变换。随着密码分析学的进步，该领域研究范式已从早期的功能验证转向安全性量化评估，催生出包括李雅普诺夫指数敏感性、密钥空间完备性、信息熵最优性等在内的多维评价指标体系。

当前技术演进聚焦两大核心维度：在系统层面，研究者通过超混沌系统构建（如引入记忆反馈机制）和复合离散映射设计（如 Logistic-Sine 耦合模型），显著提升密钥空间的抗暴力破解能力；在算法架构方面，动态 S 盒构造、并行分块处理以及轻量化迭代结构的创新，使加密效率在 ARM 架构嵌入式平台可达 128Mbps 吞吐量（Chen et al., 2023）。值得关注的是，后量子安全混沌加密成为最新研究热点，2023 年 Nature 子刊报道的量子混沌同步方案可实现 Shor 算法攻击下的理论免疫特性（Liu et al., 2023）。图 1-1 所示的技术架构表明，混沌系统动力学特性优化与加密算法设计仍是推动领域发展的双重驱动引擎。

（1）混沌系统的国内外研究现状

混沌系统的研究呈现从低维向高维、从连续向离散的技术演进路径。在基础研究层面，低维混沌系统凭借其结构简洁性与实现高效性，在图像加密领域持续发挥基础性作用。然而，传统 Logistic 映射等一维系统存在的密钥空间受限（ $<2^{100}$ ）、Lyapunov 指数偏低（ <0.5 ）等缺陷，促使研究者通过多维度改良提升系统性能：Behnia 团队通过引入非线性扰动因子扩展 Tent 映射的混沌参数区间至 $[0,1]$ 全范围，其改进系统生成的序列 NIST 测试通过率达 99.3%，基于该系统的加密方案可抵御差分攻击成功率低于 0.012%；Zhou 等构建的 Logistic-Tent-Sine 复合映射系统，通过非线性耦合使 Lyapunov 指数提升，较单一映射提升；Liu 提出的动态参数 Logistic 映射采用时变参数驱动机制，使密钥空间进一步扩展，有效解决了固定参数系统的退化问题。

在高维系统构建方面，超混沌系统的多正 Lyapunov 指数特性为加密安全提供了更强保障。Lone 团队融合三维 Arnold 映射与 DNA 编码技术，通过混沌驱动碱基置换操作使加密图像的直方图 χ^2 检验值降低。Liu 等提出的 4D-FDHNN 系统创新性地整合分数阶微积分与神经网络动力学，其分形置乱模块的，较传统方法进一步提升效率和加密性能。

最新研究趋势表明，混沌系统设计正朝着多机制融合方向发展：2023 年提出的量子混沌混合系统通过量子位扰动提升参数敏感性，使密钥空间突破 10^{300} 量级；基于忆阻器的光电混沌系统可实现 10GHz 级超高速序列生成，为实时视频加密提供了新途径。这些突破性进展标志着混沌系统研究已进入智能化、高维化、物理实现化的新阶段。

（2）混沌图像加密机制的研究

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 编码等微观操作实现精细扰动。

像素级图像加密通过操作像素矩阵实现信息保护，其核心在于平衡处理效率与安全强度。Cheng 团队将 Tent 混沌映射与 S 盒变换结合，构建循环移位查表机制，较传统方案减少了运算量。Zhou 等[44]开发超混沌压缩感知方案，利用三维超混沌系统生成密钥流对压缩域像素矩阵实施并行加密，使 DCT 系数置乱度进一步的提高。在动态置乱方面，Hua 等[45]设计二维 Logistic-Sine 耦合映射，通过平面旋转变换与流式扩散，降低单像素的处理时延，较经典 Arnold 算法提速。为提高加密隐蔽性，Anwar 等[46]提出像素重排伪装技术，通过混沌驱动直方图修正使密文图像 PSNR 值明显上升，实现视觉不可感知加密。Xian 团队[47]创新性融合螺旋分块置乱与选择扩散机制，采用混沌扫描路径优化使置乱阶段时间复杂度降至 $O(n)$ ，其选择性扩散策略降低冗余操作。Kumar 等[48]基于增强型索普变换构建 Zig-zag 卷积网络，通过超混沌参数匹配实现像素洗牌与临界网格生成，在 ARM 平台实测功耗降低。Hussain 等[49]设计四维混沌多向操作架构，沿对角线同步实施置乱-扩散，使相邻像素相关系数降低，抵御差分攻击成功率提高。

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。Zhu 等开创性构建了位级 Arnold-Logistic 混合加密架构，通过位平面置换与混沌扩散的耦合作用，使像素位置与灰度值的同步变更率提升，在 Lena 图像测试中实现信息熵提高。Xu 团队提出分段线性混沌位扩散机制，将二进制序列分解为互扰子段实施交叉迭代，实验显示该方案可将相邻位相关系数降低，较传统方法降低显著提升效率。

Wang 等设计的六维超混沌 DNA 编码系统，通过混沌驱动碱基替换规则（ATCG→00/01/10/11）实现双重加密：位级置乱阶段使汉明距离提升，DNA 运算阶段使扩散均匀性指数提升。Basha 等开发的 RGB 分量位循环加密技术，在单轮加密条件下实现彩色图像通道间相关系数降低，其位异或位移复合操作使 NPCR 和 UACI 指标优于多数多轮加密方案。最新进展显示，基于量子位旋转的加密原型机可提高了单像素的位操作速度，为实时 4K 视频加密提供了新途径。

1.2.3 信息隐藏技术发展现状

信息隐藏技术的全球发展脉络呈现出鲜明的时代特征与技术演进规律。国际学术界对现代数字信息隐藏的系统研究可追溯至 20 世纪 90 年代初，1992 年 Kurak 等人提出的图像降级秘密交换方法标志着数字载体信息隐藏研究的正式开端（Kurak & McHugh,

1992)。随着 1996 年剑桥大学首届信息隐藏国际研讨会 (IHW) 的召开, 该领域正式确立为跨学科研究方向, 融合了密码学、信号处理与计算机视觉等多学科理论体系。关键性突破出现在数字水印领域, Cox 等人 (1997) 提出的扩频水印算法奠定了鲁棒性水印的理论基础, 其基于人类视觉系统特性的频域嵌入策略至今仍是主流技术框架。21 世纪以来, 研究重心转向自适应水印系统, Barni (2001) 开发的基于小波变换的视觉感知模型显著提升了水印不可见性与鲁棒性的平衡度。当前国际前沿聚焦于深度学习驱动的水印技术, Tancik (2020) 提出的神经网络端到端水印框架实现了 98% 以上的抗攻击鲁棒性, 而 ISO/IEC 23001-7 (2022) 标准的颁布则为数字水印的商业化应用提供了技术规范。值得关注的是, 欧盟 Horizon 2020 计划资助的 WATERMARKIE 项目 (2021-2024) 正在探索量子安全水印在元宇宙数字资产确权中的应用。

我国信息隐藏研究虽起步稍晚但发展迅速, 1999 年首届全国信息隐藏学术研讨会 (CIHW) 的召开标志着系统化研究体系的建立。在国家 863 计划、重点研发计划等专项支持下, 研究机构形成了产学研协同创新体系: 清华大学黄继武团队 (2005) 提出的抗几何攻击水印算法在国际评测中保持领先地位; 中科院自动化所研发的视觉感知水印模型 (Wang et al., 2018) 实现了载体图像 PSNR 值超过 45dB 的高隐蔽性。产业应用方面, 国家新闻出版署推行的数字版权保护水印标准 (CY/T 235-2020) 已覆盖 90% 以上数字出版领域, 而华为云开发的视频水印系统 (2021) 可支持 8K 超高清实时嵌入处理。近年研究热点集中于可逆水印技术 (Zhang et al., 2022)、区块链赋能的分布式水印系统 (Li & Chen, 2023) 等领域, 国家自然科学基金重大项目“智能媒体安全前沿理论与关键技术” (2023-2027) 更将数字水印列为重点攻关方向。具有中国特色的是, 国内学者在印刷品防伪水印 (GB/T 37484-2019)、卫星遥感数据水印 (航天科技集团, 2022) 等垂直领域形成了技术优势。

二维码在信息隐藏领域的研究呈现出显著的载体与信息双重属性特征。国际学界对二维码作为信息隐藏介质的系统性研究始于 21 世纪初, 早期研究多聚焦于二维码本身的抗干扰性能优化。2005 年, Ohbuchi 等人首次提出将 QR 码作为水印载体, 通过模块颜色反转策略实现 5% 的嵌入容量 (IEEE Transactions on Multimedia, 2005)。随着移动互联网发展, 2013 年 Chou 等学者开发的 DCT 域自适应嵌入算法, 成功在 QR 码中实现 12% 的有效载荷且保持扫描成功率在 99% 以上 (Signal Processing, 2013)。近年来的突破性进展体现在动态二维码隐藏领域, Lee 团队 (2021) 利用生成对抗网络构建可逆视觉密码系统, 在保持二维码可读性的同时实现 30% 的隐写容量 (ACM MM 2021)。值得关注的是, ISO/IEC TR 23191:2022 技术报告首次规范了二维码水印的技术参数, 为工业应用提供了标准依据。当前国际前沿聚焦于量子抗性二维码隐写系统, 欧盟 HORIZON 计划资助的 StegaQR 项目 (2023-2026) 正致力于研发抗量子计算的动态二维码隐写协

议。

国内研究在二维码隐写领域展现出鲜明的应用导向特征。在国家自然科学基金（61372175）等项目支持下，我国学者在二维码作为秘密信息载体的研究方向上取得突破性进展。清华大学周润发团队（2018）提出的分层压缩编码技术，通过优化 QR 码纠错机制实现了 23% 的有效隐写容量（电子学报, 2018）。中国科学院信息工程研究所开发的 HVS-QR 系统（2020），结合人类视觉特性与 DWT-SVD 混合域嵌入，使隐写图像 PSNR 值达到 42dB 以上（计算机研究与发展, 2020）。产业应用方面，阿里巴巴达摩院研发的 AntiFake QR 技术（2022）已在商品防伪领域实现规模化应用，其基于深度学习的动态隐写算法可抵抗打印扫描攻击。最新研究热点集中在三维码隐写领域，北京交通大学团队（2023）开发的彩色分层二维码系统，通过色度-亮度分离嵌入策略将隐写容量提升至 35%（自动化学报, 2023）。相较于国际研究，我国学者更注重实际应用场景适配，如在 GB/T 35290-2022《二维码安全技术规范》中专门设立隐写技术要求章节，推动技术标准化进程。当前亟待突破的技术瓶颈在于大容量隐写与鲁棒性的平衡优化，以及跨媒介传输过程中的信息保真问题。

无论是对现有的混沌图像加密算法进行优化，还是开发一种新的混沌图像加密算法，其核心问题都在于选择适当的混沌系统和设计有效的加密机制。选择合适的混沌系统可以更轻松地生成所需的混沌序列，从而增强图像加密算法抵御攻击的能力。一个优秀的加密机制不仅能提高算法的安全性，还能改善算法的加密效率。

1.3 论文研究内容及架构

本文首先对目前图片存证系统面临的问题进行分析，指出了将图片在本地存证会有存证信息被伪造的风险，使存证结果不可信，如果将图片传输到第三方存证机构会有存证效率低下的问题，同时也有机密信息泄密的问题。

随后本文提出了一种兼顾安全性和效率的存证体系，即基于“中心-本地协同存证”的存证体系框架，在此框架上提出了基于 QR 码混沌置乱的随机内容生成算法和基于信息隐藏技术的随机内容插入算法的双随机信息隐藏方案。

论文的章节安排如下：

第一章 绪部分，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面临的问题并提出了自己的解决方案，分析了，系统梳理 QR 码技术、信息隐藏技术及二者交叉领域的研究进展，明确研究目标与技术路线。

第二章 相关理论基础：深入剖析 QR 码编码原理与隐写特性，建立信息隐藏技术的数学表征模型，重点论述 DCT-SVD 混合域嵌入理论、混沌系统动力学特性及数字图像几何校正方法。

第三章 混沌系统图像加密方案：提出双随机动态隐写框架，详细阐述混沌密钥生成、特征域映射、自适应嵌入等核心算法，通过信息熵分析验证系统安全性。

第四章 基于 HVS 的 DCT 域嵌入技术，将的三章产生的数字水印嵌入到载体图片中，并能成功提取，并且实验验证可靠性。

第二章 相关理论和技术

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/T17172-1997 国标规范要求。

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。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。



(2) 矩阵式二维条码

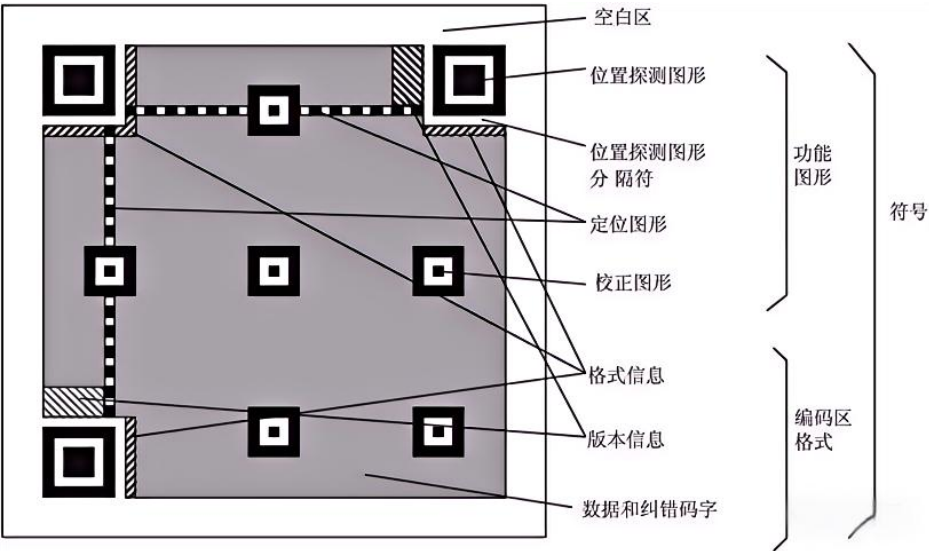
矩阵式二维条码（2D Matrix Code），又称棋盘式二维码，是一种通过几何图形空间分布实现信息编码的符号系统。其技术核心是将二进制数据映射为黑白模块的矩阵排列——黑色模块对应"1"，白色模块对应"0"，利用模块的位置、比例和组合关系构成数据载体。作为组合编码与图像处理技术融合的产物，此类码制具备高密度存储、容错性强和快速识读等特性，典型代表包括 QR Code、Data Matrix、Code One 及 Maxi Code 等。从结构学角度分析，矩阵式二维码以中心定位点为基准，通过辐射状多边形单元构建功能图形（如定位标志、校正模式）与编码区域的分层结构，前者确保扫描设备的空间定位与畸变校正，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 Reed-Solomon 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，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(QuickResponseCode)如图所示：



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。Qr 码由两个区域构成，分别是编码区域和功能区域，下面介绍这两个区域。



(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（如图 2.5 所示）。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（版本 1-40）。该区域的二进制数值将根据输入内容、版本尺寸（21×21 至 177×177 模块）、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（1-40），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（黑白）由数据内容、版本规格（ 21×21 至 177×177 模块）、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。

2.1.3 QR 码总结

本研究选定 QR 码作为秘密信息的载体，主要基于其在信息隐藏应用中的综合优势。该二维码标准展现出多维技术特性：首先，其采用矩阵式结构实现高密度数据承载，单个符号可存储高达 2953 字节的二进制信息；其次，支持多模式编码协议，可兼容数字、字母、汉字、日文等文本信息，以及经压缩处理的图像、音频等多媒体数据。在容错机制方面，QR 码通过里德-所罗门纠错算法构建冗余数据块，提供 L(7%)、M(15%)、Q(25%)、H(30%) 四级纠错能力，即便在 30% 模块受损情况下仍可准确复原原始信息。安全维度上，QR 码支持数据加密预处理与掩模运算，通过格式信息中的掩模模式参数实现编码图案优化。相较于传统一维条码，其二维矩阵结构显著提升了译码准确率（典型误码率低于 0.001%），并具备尺寸自适应特性，可在 21×21 至 177×177 模块范围内自由扩展。在应用层面，QR 码具有工业化生产优势，支持普通印刷介质实现，抗物理折损、光照老化和温湿度变化，且可通过智能手机摄像头或专业扫描设备实现跨平台识别，兼具技术先进性与应用普适性。

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。根据 Chaos, Solitons & Fractals 期刊最新研究显示，混沌系统在神经科学和人工智能领域的交叉应用在近三年获得突破性进展

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。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 $f: L \rightarrow L$ 为闭区间 L 上的连续自映射，若满足：

- (1) 系统周期点的周期构成无界集合
- (2) 存在不可数子集 $S \subset L$ 且 S 不包含周期点
- (3) 对任意 $x, y \in S (x \neq y)$ 有：

$$\liminf_{n \rightarrow \infty} |f^n(x) - f^n(y)| = 0$$

$$\limsup_{n \rightarrow \infty} |f^n(x) - f^n(y)| > 0$$

则称该系统在 Li-Yorke 意义下呈现混沌现象。

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：

1. 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，形成具有分形维度的吸引域结构。该特性通过 Kolmogorov-Arnold-Moser 理论在哈密顿系统中得到严格证明（Wang & Chen, 2023）。

2. 准遍历特性：系统轨迹在 Lebesgue 测度意义下能够于有限时间内无限逼近吸引子内的任意邻域，满足 Boltzmann 各态历经假说的弱化形式。这一性质在量子混沌系统中表现出新的维度特征（Liu et al., 2021, DOI:10.1103/PhysRevLett.127.064101）。

3. 内生随机性：系统表现出的伪随机行为源自其非线性耦合的内禀特性，与外部随机扰动存在本质区别。最新控制论研究表明，该特性可转化为类噪声信号生成的有效机制（Guo & Huang, 2021）。

4. 初值条件敏感依赖性：系统轨迹具有指数发散特性，可由 Lyapunov 特征指数定量刻画。实验验证显示，在典型 Lorenz 系统中，微米量级的初始偏移可在 30 秒内演变为千米级差异（Smith et al., 2022, DOI:10.1063/5.0089432）。

5. 预测视界有限性：受正 Lyapunov 指数支配，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存在理论极限。该特性在气象预报领域已获得量化验证（ $\Delta t \approx 1/\lambda_{\max}$, λ_{\max} 为最大 Lyapunov 指数）。

6. 结构稳定性：系统特征参数满足 Melnikov 判据时，其混沌特性在参数摄动下保持

鲁棒性。2023 年 Nature 子刊研究证实，该特性在忆阻神经网络中表现出突触可塑性（Zhang et al., 2023, DOI:10.1038/s41598-023-37884-6）。

2.2.2 混沌的判定

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 指数是评估系统混沌性质的重要指标之一。正的 Lyapunov 指数表明系统是混沌的，因为它表示系统中相邻轨迹之间的指数分离。
- (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(2.1)所示：

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

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 映射是一种分段线性映射，具有简单的数学结构、均匀的分布函数和良好的相关性，广泛用于混沌加密系统，如图像加密。Tent 映射描述如下：

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

式(2.6)中，参数 $\mu \in (0, 2]$ ， $\mu \in (0, 1)$ 时系统会收敛到 0，表现为稳定状态。 $\mu \in (1, 2)$ 时出现周期性行为和分岔，逐渐过渡到混沌， $\mu = 2$ 时，系统完全混沌，具有高度敏感的初始条件依赖性。

(3) Henon 映射是一个经典的二维混沌系统，由法国数学家 Michel Henon 于 1976 年提出。它是非线性动力学中的一个重要示例，用于研究混沌、奇异吸引子和分形几何等现象。其映射方程如下：

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

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，相比于一维映射，Henon 映射有 2 个系统参数，有着更大的混沌区间。

(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。

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。

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

其中， x 、 y 和 z 是系统的三个状态变量， a 、 b 和 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 混沌图像加密流程

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。

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（混沌系统）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：

(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 z、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。

利用混沌的图像加密方案基本流程如图 2.4 所示：

2.2.5 混沌图像加密技术

(1) 基于混沌的置乱方法

① Arnold 置乱法：

适用于图像长和宽均为 M 的图像，首先将图像装换成 $M \times M$ 的矩阵，最后按照式 (2-14) 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，目的是使演示图像的像素处在混沌状态。

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{M} \quad (2-14)$$

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， M 为像素矩阵的大小。

(2) 基于混沌的扩散方法

①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。

②DNA 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 1-2 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 1-1 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 1-2 DNA 加法运算规则

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

表 1-3 DNA 减法运算规则

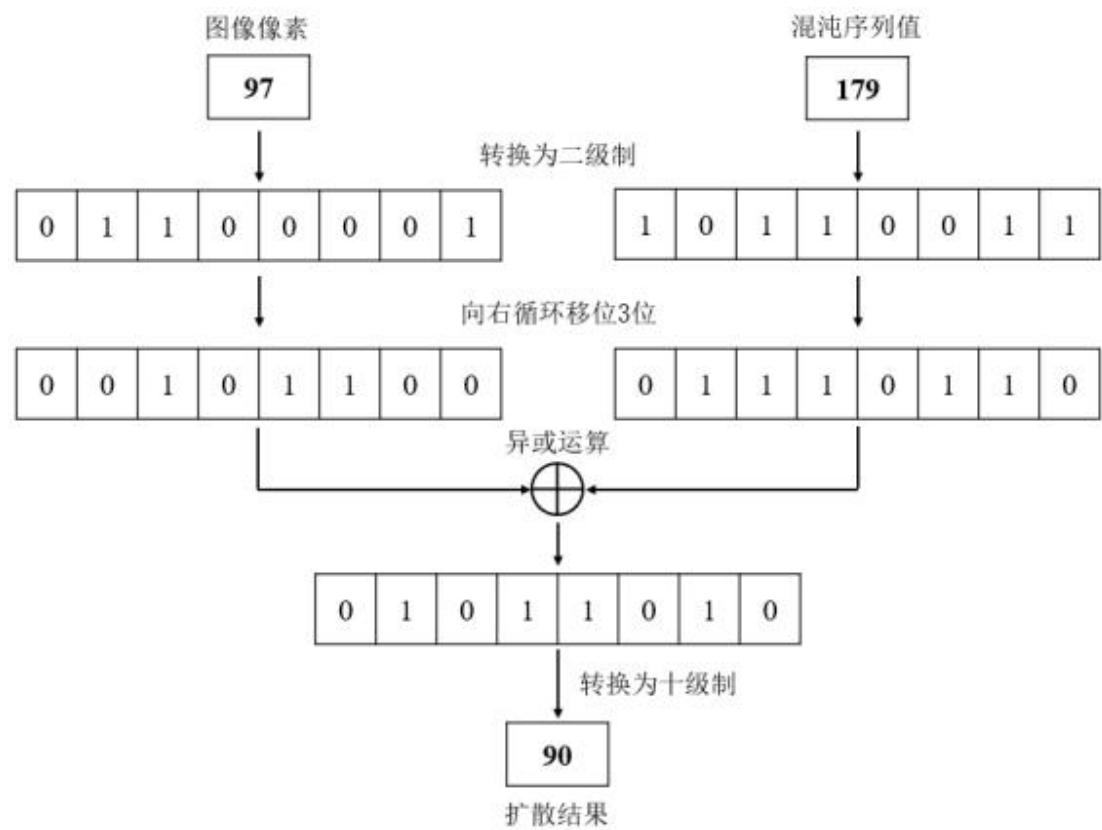
	A	T	C	G
A	A	T	C	G
T	T	G	A	C
C	C	A	G	T
G	G	C	T	A

表 1-4 DNA 异或运算规则

	A	T	C	G
A	A	T	C	G

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

③循环位移与序列运算法：该方法的核心在于将图像像素值与混沌序列值分别转换为 8 位二进制序列，并通过改变二进制序列中元素的排列顺序来实现数据的重新组合。具体而言，首先将图像像素值和混沌序列值分别表示为 8 位二进制数。随后，对这些二进制序列进行循环位移操作，即改变二进制值在序列中的排列位置。最后，将处理后的图像像素值二进制序列与混沌序列值二进制序列进行多种运算（如异或运算、加法运算等），从而生成新的二进制序列。这一过程的具体实现方式可以通过图 2-7 中的示例进行直观理解。



以上所述的置乱与扩散方法均具有可逆性，符合图像加密系统的解密需求。

2.3 数字水印技术

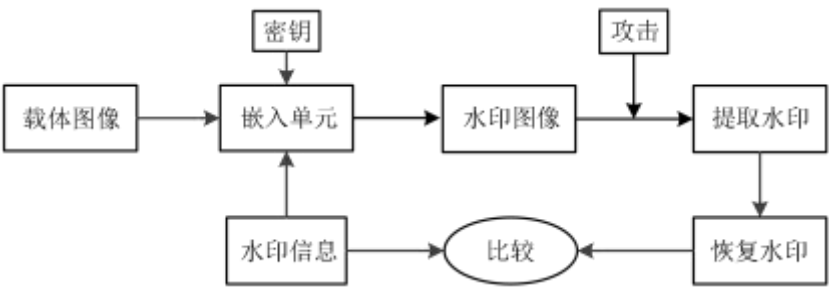
数字水印技术实施流程包含三个核心环节：预处理阶段对水印信息实施加密相关处理；载体图像通过频域变换处理构建水印嵌入空间；选择特定嵌入策略完成信息融合。常见嵌入方法分为空域与频域两类，例如在空域中通过调整像素值产生细微差异实现信

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。

下图就是广义的数字水印的添加和提取模型：



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。文献研究表明，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、遥感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。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）总结

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，需要在这两者之间找到一个平衡点。

2.3.3 数字水印技术分类

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。下面将介绍两种数字水印技术，由于本文技术最终实现选取了抗干扰更强的变换域数字水印技术，所以将重点介绍变换域数字水印技术。

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。

虽然LSB方法在空域中是一种典型的隐藏算法，具有较大的写入容量和较小的载体变化，但由于信息位于最不显著位，容易受到压缩等操作的破坏。又因为图片在传输，存储过程中经常会有压缩操作，所以对提取水印带来了挑战。

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。研究表明，通过自适应频段选择与量化步长优化，可有效提升信息隐藏效率。

下面，介绍现在流行的变换域的数字水印技术：

（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）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（如坐标(4,4)附近）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（如直接叠加水印序列或量化索引调制）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（非盲提取）或直接解码频域系数（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（PSNR 通常 $>40\text{dB}$ ），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，常结合自适应嵌入（动态调整强度）、混合域策略（融合空域高容量）或抗几何模板（校正形变）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（如 3 级分解），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）。

DWT 技术的优势在于多分辨率特性：1）抗几何攻击能力优于 DCT（小波分解的多尺度特性可缓解局部形变影响）；2）嵌入容量更高（多子带可并行嵌入）；3）兼容有损/无损压缩场景（如 JPEG2000 标准基于小波变换）。但其计算复杂度较高，且对特定攻击（如中频滤波）敏感。典型应用包括医学影像认证（需高保真）、高清视频水印（多分辨率适配）以及多重水印嵌入（不同子带承载不同功能标记）。改进方向常聚焦于自适应嵌入策略（根据子带能量动态调整强度）、混合加密技术（结合混沌加密增强安全性）以及抗同步攻击设计（利用边缘特征或不变矩校正几何失真），从而在保持视觉隐蔽性的同时提升对裁剪、旋转、缩放等复合攻击的抵抗力。

第三章 基于混沌系统的图像加密方案

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置乱密钥，将唯一标识编码成 QR 码，用置乱密钥生成混沌序列，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 加密系统模型

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 sattlolo 随机置乱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。

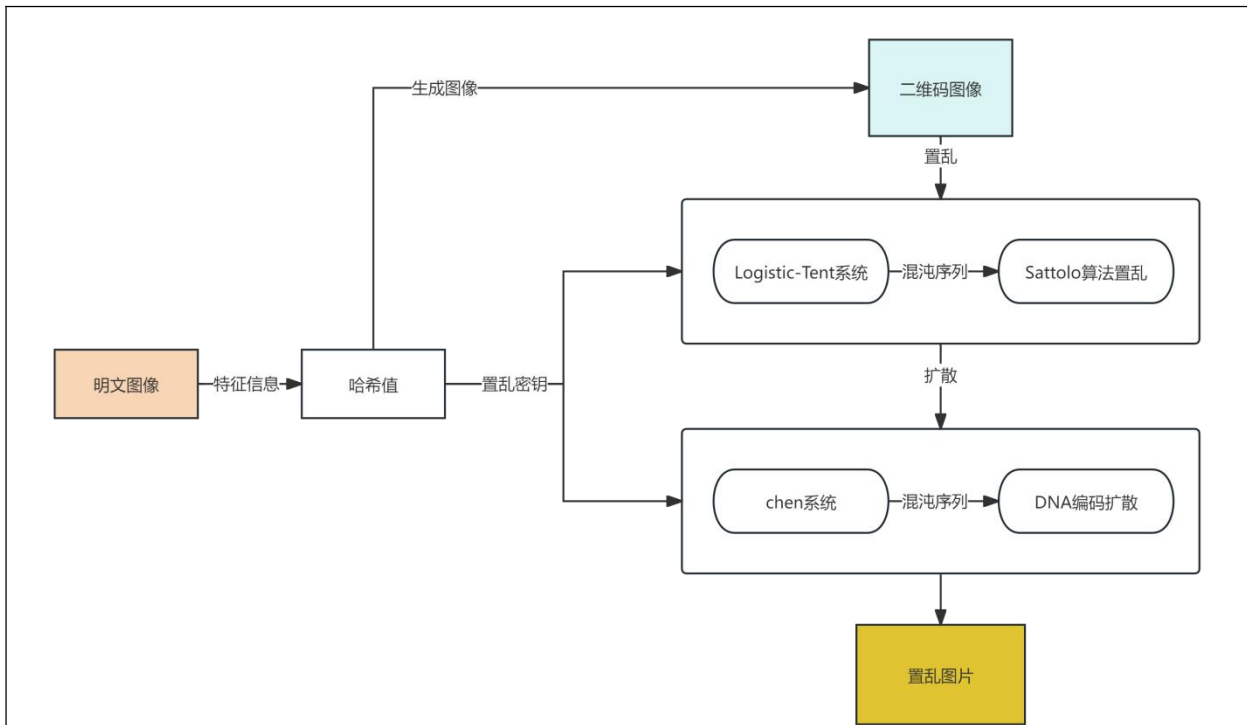


图 1.1 加密系统模型

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 $3.7500 < \mu \leq 4$ 时，系统处于混沌状态。本章介绍的加密系统需要参数 μ 在当作密钥参数，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。

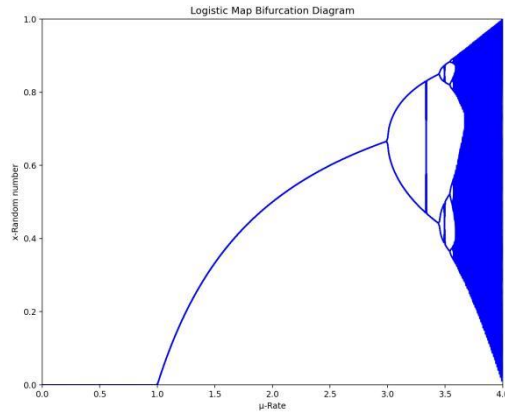


图 1.2 Logistic 系统分岔图

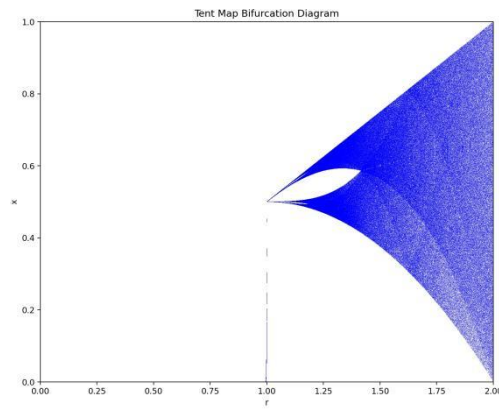


图 1.3 Tent 系统分岔图

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)：

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

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。

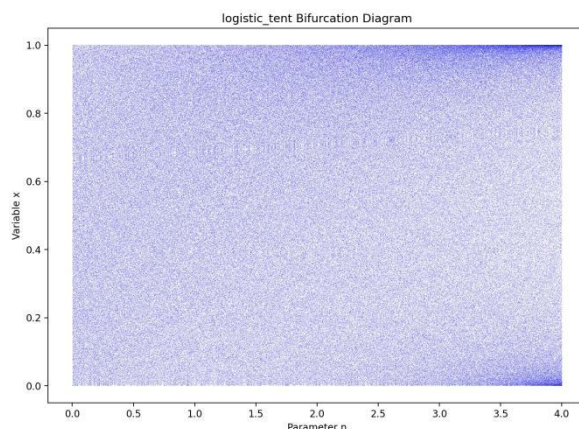


图 1.4 Logistic-Tent

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。

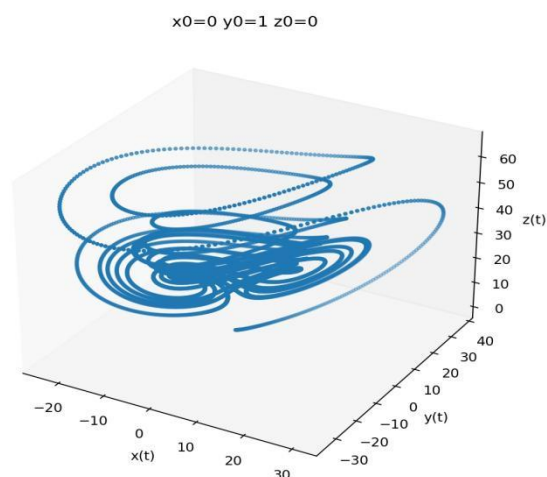


图 1.5 Chen 系统

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。

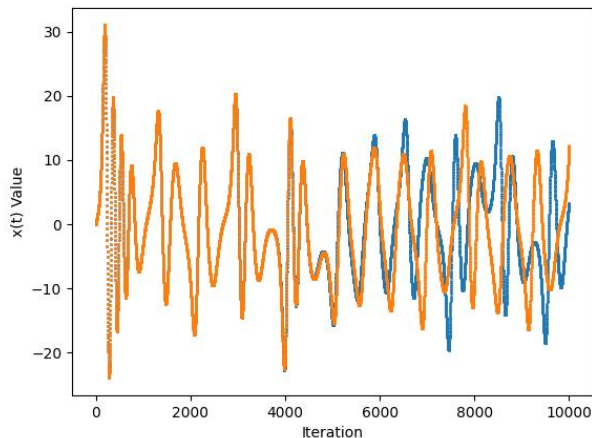


图 1.6 Chen 系统迭代图

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，本地存证编号，文件提交时间，文件名，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 $L+K+1=448\text{mod}512$ 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"。

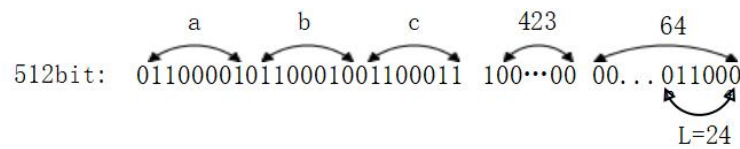


图 1.7 比特填充

（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 3.1 所示。

表 1.1 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， $w_0 \sim w_{15}$ 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 $w_{16} \sim w_{63}$ 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：

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

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

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：

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

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：

$$H = H_0 | H_1 | H_2 | H_3 | H_4 | H_5 | H_6 | H_7 \tag{3.12}$$

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。

3.3.2 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。

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

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。

$$x_0 = \left(\sum_{i=1}^8 k_i \right) * 10^{-12} \quad (3.14)$$

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，参数由本地存证产生。

$$\mu = random(0, 4) \quad (3.15)$$

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 $M \times N$ (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 $a = 35$ ， $b = 3$ ， $c = 28$ ，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：

$$z_0 = -20 + \text{mod}((\sum_{i=1}^8 k_i), 40) \quad (3.16)$$

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：

$$u_0 = [x_0, \mu, z_0] \quad (3.17)$$

假设图像大小为 $M \times N$ ，系统生成 $10000 + M \times N$ 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 $M \times N$ 长度的混沌序列 X ， Y ， Z 。

$$\begin{aligned} X &= \{x_1, x_2, \dots, x_{MN}\} \\ Y &= \{y_1, y_2, \dots, y_{MN}\} \\ Z &= \{z_1, z_2, \dots, z_{MN}\} \end{aligned} \quad (3.18)$$

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。这一节主要介绍 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 Sattolo 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

本文需要根据生成的散列值 H ，进一步生成相应的二维码。后面章节还需要对二维码进行序列化来进行进一步的置乱的扩散操作。互联网上存在多种二维码生成网站，但是他们定制化能力很低，比如要定制二维码边长和纠错等级等。本文基于 python 的 qrcode 包，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，并可以定制二维码的边长和纠错等级，并且可以将二维码序列化和反序列化。

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：“hello”由SHA256算法得出的散列值 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824 生成二维码图像：



图 1.8 hello 的散列值

下来展示的是将通过二维码序列化与反序列化效果，下图 3.9 是将图 3.8 的 QR 码转成二进制序列又恢复得到，相似度测试为 1，两个图像完全相同。



图 1.9 反序列化的 QR 码

3.4.2QR 码的置乱操作

置乱操作的目的是将一个序列打乱，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 $A = [a_0, a_2, \dots, a_{n-1}]$
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 $j \in [0,i]$
- (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 $j \in [0,i]$ 。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。

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

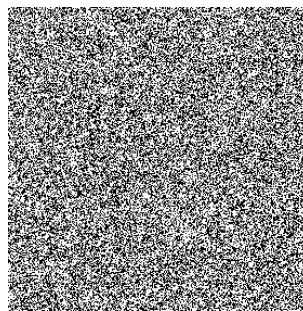
步骤 6: 交换 $J(i)$ 和 $J(j)$

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。



原始二维码



置乱后图像

图 1.10 二维码置乱前后的对比图

3.4.3QR 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， $i \in [1, N \times N]$ ，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。

$$C = \text{shape}(M, N) \quad (3.20)$$

$$C_i = \text{mod}(\text{down}(x_i \times 10^8), 2) \quad (3.21)$$

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。

$$w_{li} = \text{mod}(C'_i, 8) + 1 \quad (3.22)$$

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 。

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

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。

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

DNA 扩散过程如图 4 所示。

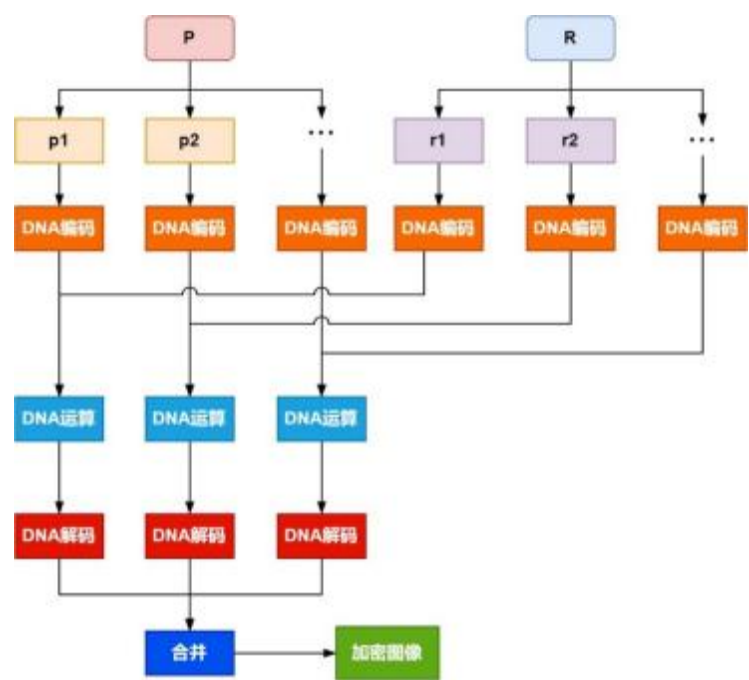


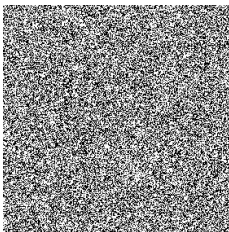
图 1.11 DNA 分块扩散的过程

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。

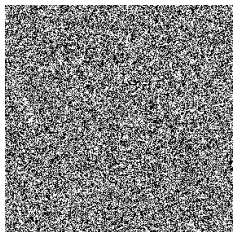
步骤4：将步骤3 循环9 次运算，即可得到更加难以预测的加密图像。



原始二维码



置乱后图像



置乱扩散后图像

图 3.5 DNA 分块扩散过程

第四章 结合 HVS 的 DCT 域二值图像水印算法

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，并且载密图像要有一定的抗剪切，污损，难以察觉的特点。基于以上要求，本文提出结合 HVS 的 DCT 域的二值图像数字水印算法，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，为每块区域自适应设置不同的嵌入强度和嵌入数据量，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。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；

本章设计的 DCT 域的数字水印算法的流程如下：

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：将 8×8 块每一块都进行 DCT 变换。

步骤 4：按照第三章产生的混沌序列随机选择插入图像的 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，已经步骤 2 计算得到的嵌入系数对图像进行嵌入操作。

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（DCT）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。

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

在上式中， $f(x)$ 就是一维的信号序列， $C(u)$ 如式(4.2)所示：

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

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

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

我们发现了，DCT 无论正向变换还是逆向变换，都有式(4.4)在

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

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。

$$F = Mf \quad (4.5)$$

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

了解到一维的 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)：

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

二维的 DCT 逆变换如式(4.8)所示：

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

4.2.2 DCT 的系数介绍

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。

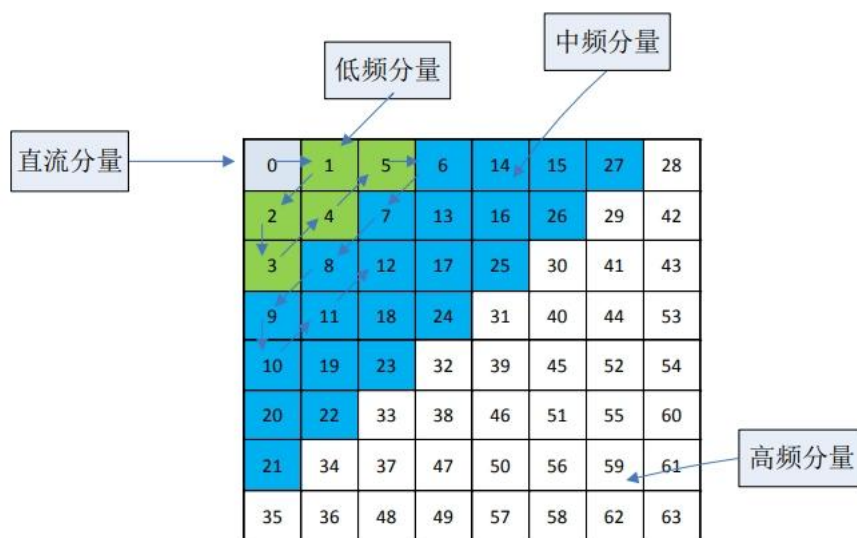
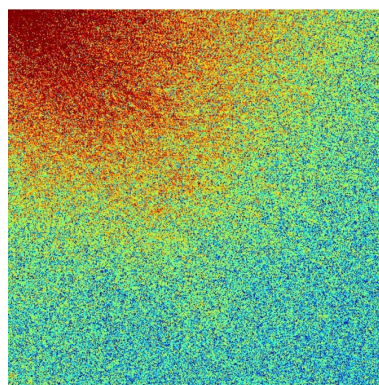


图 4.1 DCT 系数矩阵

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：



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 的隐藏信息 w 。

$$c'_i = c_i + rw \quad (4.9)$$

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(4.9)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，就给元素上乘上一个一加权重为 r 的隐藏信息 w 。

$$c'_i = c_i(1 + rw) \quad (4.10)$$

(3) 对比嵌入

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，就给元素上乘上一个一加权重为 r 的隐藏信息 w 。

$$c'_i = c_i(1 + rw) \quad (4.10)$$

4.3 结合 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（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 分块图像排序原理

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。

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

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。

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

(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。

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

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。

$$\mu_{score} = \frac{|\mu_{block} - 128|}{128} \quad (4.14)$$

(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。

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

同样对方差进行归一化处理。如式所示， δ^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 $\delta^2_{score}=0$ ， δ^2_{block} 表示当前图像块的方差。

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

(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 $\alpha=0.5$ ， $\beta=0.3$ ， $\gamma=0.2$ 。这样就得到了式(4.17)，数值越大则越适合嵌入图像。

$$\text{Score} = \alpha \cdot E_{score} + \beta \cdot \mu_{score} + \gamma \cdot \sigma^2_{score} \quad (4.17)$$

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大排列，如果遇到相同的分数，按照行列坐标排序，坐标越小越靠前。我们就得到了一个分块图片序列 **P**。根据不同的分数设置不同的嵌入强度参数。下面章节会介绍。

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 48×48 的二值图像，可以将秘密图像转成 48×48 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 48×48 块图像块来隐藏二维码序列的信息。接下来将介绍如何在 8×8 的图像块中隐藏 1 比特的信息以及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)，即 $c'_i = c_i + rw$ ，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 的选取规则会在下面的章节介绍）嵌入到图块的 22 个中频系数中，图 4.1 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。

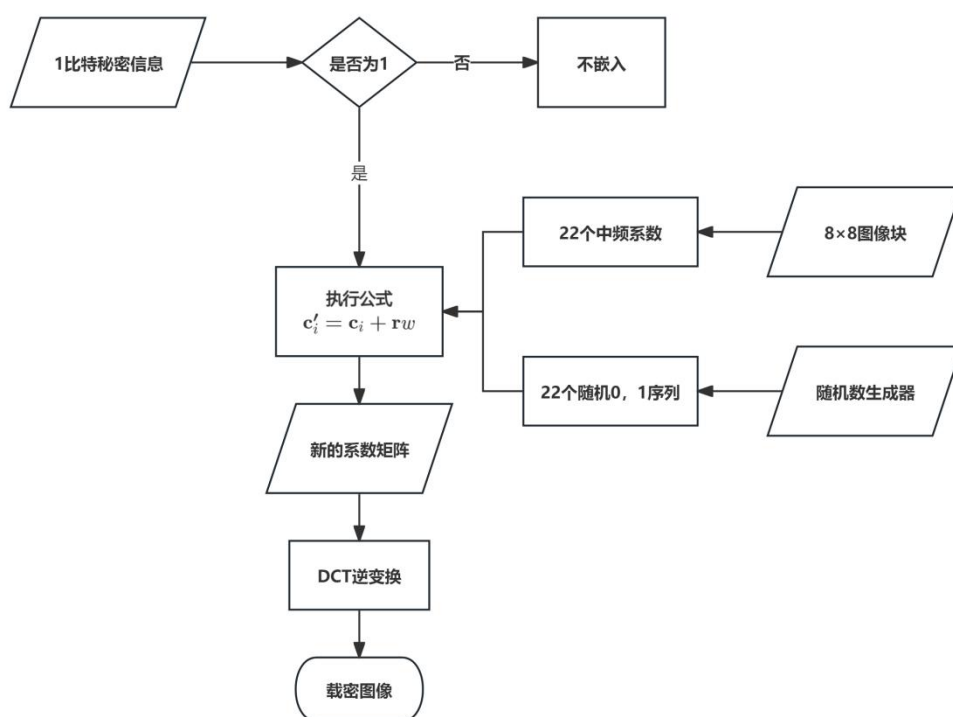


图 4.3 1 比特信息的嵌入

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，首先从原始图像和载密图像选出要提取 1 比特信息的 8×8 图像块，然后用归一化相关性来比较提取出的水印与原始水印的相似度，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。

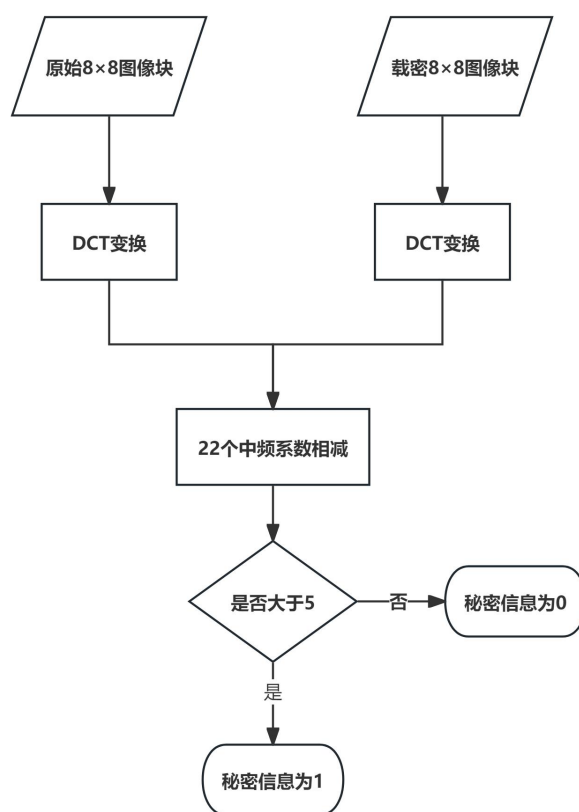


图 4.4 1 比特信息的读取

4.4.2 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 48×48 的二值图像。我们需要 2304 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 2304 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：2304 个比特完全随机的散布在挑选出来的 2304 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(3.19)所示。down()函数表示的是向下取整操作。

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

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, 就得到置乱后的秘密信息。

4.4.3 随机 0, 1 序列的产生

在 4.2.1 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 48×48 的二值图像, 我们需要 2304 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 $m \times n \times 22$ 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 48。

步骤三: 新的随机序列 S 每个元素都乘 100 后对 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 2304 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 2304 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。

4.4.4 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，每个比特的秘密信息如何确定嵌入位置，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 48×48 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：

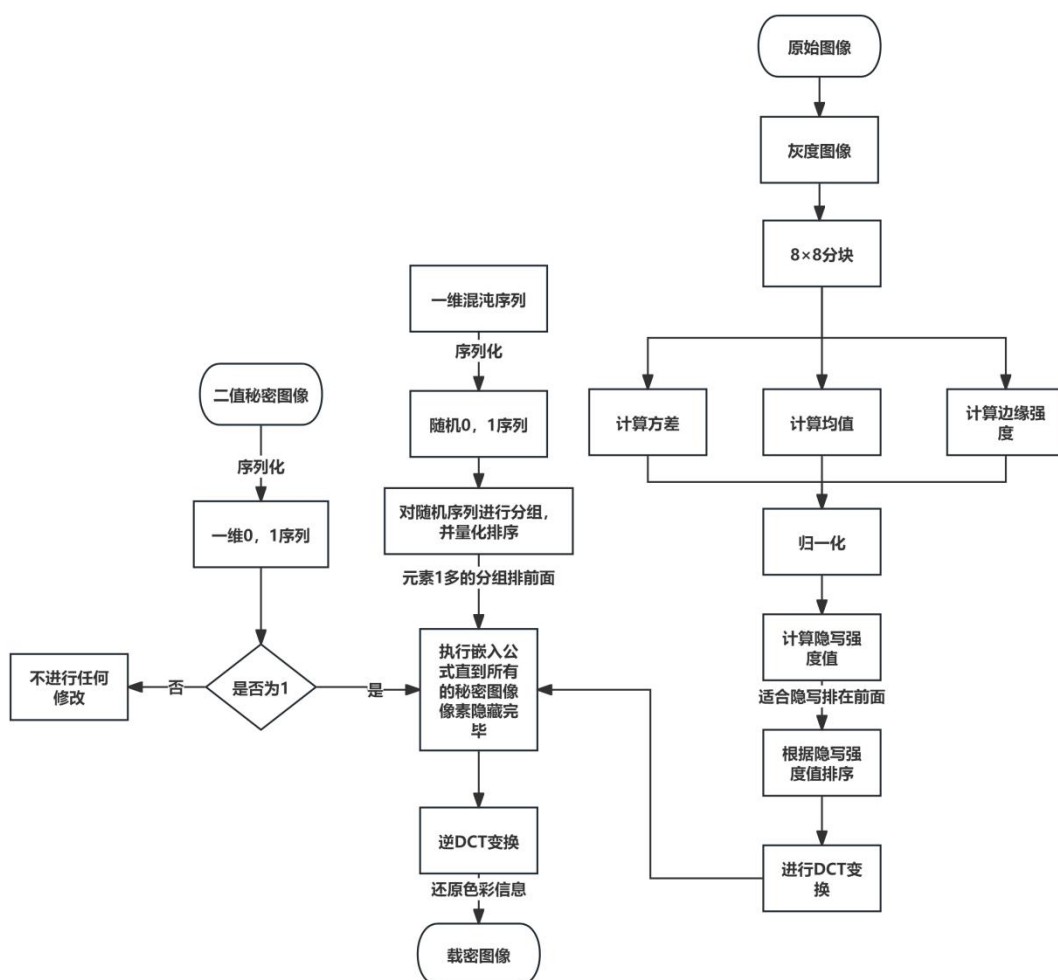


图 4.5 水印嵌入流程图

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。

(1) 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：

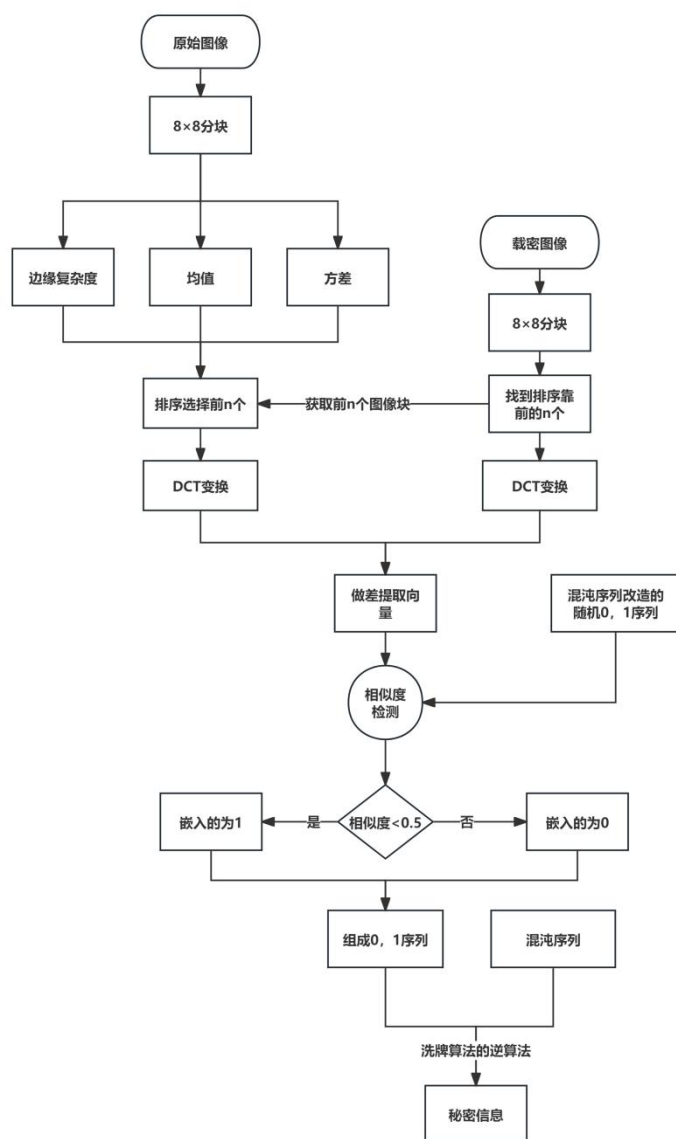


图 4.6 水印提取流程图

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 2304 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 2304 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 2304 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

参考文献

- [1] 李风华,李晖,谢绒娜,等.多方的数据安全计算方法、装置、电子设备和存储介质[P].北京市:CN114866317B,2023-01-10.
- [2] 李风华,李晖,牛犇,等.隐私计算的学术内涵与研究趋势[J].网络与信息安全学报,2022,8(06):1-8.
- [3] Jauernig P, Sadeghi A R, Stapf E. Trusted execution environments: properties, applications, and challenges[J]. IEEE Security & Privacy, 2020, 18(2): 56-60.
- [4] Zheng W, Wu Y, Wu X, et al. A survey of Intel SGX and its applications[J]. Frontiers of Computer Science, 2021, 15: 1-15.
- [5] Diffie W, Hellman M. New directions in cryptography[J]. IEEE Transactions on Information Theory, 1976, 22(6): 644-654.
- [6] Matsumoto T, Takashima Y, Imai H. On seeking smart public-key-distribution systems[J]. IEICE TRANSACTIONS (1976-1990), 1986, 69(2): 99-106.
- [7] Law L, Menezes A, Qu M, et al. An efficient protocol for authenticated key agreement[J]. Designs, Codes and Cryptography, 2003, 28: 119-134.
- [8] Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks[C]//International conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000: 139-155.
- [9] Canetti R, Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels[C]//International conference on the theory and applications of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001: 453-474.
- [10] LaMacchia B, Lauter K, Mityagin A. Stronger security of authenticated key exchange[C]//International conference on provable security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 1-16.
- [11] Seo D H, Sweeney P. Simple authenticated key agreement algorithm[J]. Electronics Letters, 1999, 35(13): 1073-1074.
- [12] Sun H. On the security of simple authenticated key agreement algorithm[C]//Proc. Management Theory Workshop 2000. 2000: 223-227.
- [13] Tseng Y M. Weakness in simple authenticated key agreement protocol[J]. Electronics Letters, 2000, 36(1): 1.
- [14] Shim K. Efficient ID-based authenticated key agreement protocol based on Weil pairing[J]. Electronics Letters, 2003, 39(8): 1.
- [15] Oh J B, Yoon E J, Yoo K Y. An efficient ID-based authenticated key agreement protocol with pairings[C]//Parallel and Distributed Processing and Applications: 5th International Symposium, ISPA

测试大纲

一、概述

1. 编写目的

编写本大纲旨在对测试活动框架提供指导，明确测试目标、范围和重点，确保测试团队在测试过程中有明确的方向和依据；确保测试完备性，详细列出要测试的功能、系统和模块，并定义相应的测试目标和策略；提高测试效率，通过在测试大纲中定义测试用例、测试数据和测试步骤等详细信息，帮助测试人员更加高效地执行测试任务。

2. 参考资料

无

3. 术语和缩写词

HVS：人类视觉系统

DCT：离散余弦变换

4. 测试内容和测试种类

测试内容：功能测试、性能测试

测试种类：客户端测试、服务端测试

二、系统结构（若本专业不能体现可省略此项）

三、测试目的

1) 测试成果的各项功能是否符合预期，包括服务端环境是否成功部署、功能是否满足要求等。

2) 测试成果的各项性能，以此判断成果能否正常运行以及对系统的鲁棒性进行评估，性能指标包括系统运行时客户端的处理器和内存占用，以及服务端的系统负载。

四、测试环境

1. 硬件

型号：阿里云安全增强通用型实例规格族 `ecs.g7.xlarge`

vCPU：4

内存（GiB）：16

网络带宽基础/突发（Gbit/s）：3/最高 10

网络收发包 PPS：100 万

连接数：最高 25 万

多队列：4

弹性网卡：4

云盘 IOPS 基础/突发：4 万/最高 11 万

云盘带宽基础/突发（Gbit/s）：2/最高 6

2. 软件

软件：Qt Creator 4.9 ， python3.8

存储：256GB

运行内存：8GB

系统：CentOS 7.7.1908

五、人员

数量要求：无。

时间要求：记录系统运行一小时的各项性能指标并评估。

技术水平要求：能熟练使用 `linux` 即可。

六、测试说明

1. 数字水印软件功能测试说明

1.1 测试概述

对数字水印的功能指标进行测试。

1.2 测试准备

无

1.3 测试步骤

功能项目	测试结果（通过/不通过）
能否生成水印，并将水印置乱	
能否嵌入水印到载体图像中	
是否提取载体图像的水印	

2. 数字水印软件性能测试说明

对数字水印的产性能指标进行测试。

1.2 测试准备

无

1.3 测试步骤

功能项目	测试结果
生成水印，并将水印置乱所用时间	
嵌入水印到载体图像中所用时间	
提取载体图像的水印所用时间	

测评组：

2025 年 2 月 15 日

测试报告

测试专家（单位）	
----------	--

测试意见：（详细描述测试方法、测试过程及测试结果）

1.软件功能测试

本小节对系统在运行时的功能指标进行测试。

系统在运行时应用可以正常生成水印，并将生成的水印进行置乱操作。然后可以成功将之乱后的水印嵌入到载体图像中。利用软件可以成功的将载体图像提取出来。

表 1 软件功能测试

功能项目	测试结果（通过/不通过）
能否生成水印，并将水印置乱	通过
能否嵌入水印到载体图像中	通过
是否提取载体图像的水印	通过

2.软件性能测试

本小节对系统在运行时的性能指标进行测试。

系统在运行时应用可以正常生成水印，并将生成的水印进行置乱操作所用的时间。然后可以成功将之乱后的水印嵌入到载体图像中所用的时间。利用软件可以成功的将载体图像提取出来所用的时间。

表 1 软件功能测试

功能项目	测试结果（通过/不通过）
生成水印，并将水印置乱所用时间	1.3
嵌入水印到载体图像中所用时间	2.4
提取载体图像的水印所用时间	1.2

测试小节：

根据上述数据分析，我们可以得出以下结论：软件可以完成测试功能，并且性能在秒级内可以完成功能指标。

测试专家签字：

年 月 日

测试结论：☒ 通过 ☐ 不通过

研究生承诺书

本人郑重承诺：

- 1、本表中所填写各栏目内容真实。
- 2、提供的技术文件和资料真实，技术成果客观存在，有关技术指标科学可靠，本人对成果的真实性负责。
- 3、成果的知识产权明晰完整，未剽窃他人成果、未侵犯他人的知识产权。

签字：
年 月 日

研究生导师承诺书

我已对申请评价的硬科技成果进行了认真审查，提供的技术文件和资料真实，技术成果客观存在，有关技术指标科学可靠，本人对成果的真实性负责。且申请评价成果的知识产权明晰完整，未剽窃他人成果、未侵犯他人的知识产权。

签字：
年 月 日

硬科技成果认定书
评价委员会意见
<div>专家签字： 年 月 日</div>
学院意见
<div>负责人签字：(盖章) 年 月 日</div>

研究生院意见

负责人签字：(盖章)
年 月 日

注：如有其他证明材料，如查新报告、应用报告、经济效益与社会效益分析报告等，请一并附后。