

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材  
计算机科学与技术

# 信息隐藏原理及应用

葛秀慧 田浩 郭立甫 韩缇文 编著

清华大学出版社



高等学校教材  
计算机科学与技术

# 信息隐藏原理及应用

葛秀慧 田浩 郭立甫 韩缇文 编著

清华大学出版社  
北京

## 内 容 简 介

本书全面系统地论述了信息隐藏的概念、分类、应用、理论与原理。书中重点介绍了信息隐藏的基本原理，并分析了与其相关的典型算法，以丰富的实例进行说明，同时提供了部分源代码。另外还详细讨论了数字水印技术与算法，探讨了隐写分析与隐蔽通信。

本书可以作为计算机应用、网络工程、通信与信息系统、信号与处理、信息安全与密码学、电子商务专业的本科生和研究生的教材，也可供从事信息安全研究及应用的学者、技术人员参考。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

## 图书在版编目(CIP)数据

信息隐藏原理及应用 / 葛秀慧等编著. —北京：清华大学出版社，2008.10  
(高等学校教材·计算机科学与技术)

ISBN 978-7-302-18324-2

I. 信… II. 葛… III. 信息系—安全技术—高等学校—教材 IV. TP309

中国版本图书馆 CIP 数据核字(2008)第 118316 号

责任编辑：闫红梅 林都嘉

责任校对：焦丽丽

责任印制：杨 艳

出版发行：清华大学出版社 地 址：北京清华大学学研大厦 A 座

http://www.tup.com.cn 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969,c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者：北京市清华园胶印厂

经 销：全国新华书店

开 本：185×260 印 张：10.25 字 数：250 千字

版 次：2008 年 10 月第 1 版 印 次：2008 年 10 月第 1 次印刷

印 数：1~2500

定 价：18.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103 产品编号：025532-01

## 编审委员会成员

(按地区排序)

清华大学

周立柱 教授  
覃 征 教授  
王建民 教授  
刘 强 副教授  
冯建华 副教授

北京大学

杨冬青 教授  
陈 钟 教授  
陈立军 副教授

北京航空航天大学

马殿富 教授  
吴超英 副教授  
姚淑珍 教授

中国人民大学

王 珊 教授  
孟小峰 教授  
陈 红 教授

北京师范大学

周明全 教授  
阮秋琦 教授  
孟庆昌 教授

北京交通大学

杨炳儒 教授  
陈 明 教授  
艾德才 教授

北京信息工程学院

吴立德 教授  
吴百锋 教授  
杨卫东 副教授

北京科技大学

邵志清 教授  
杨宗源 教授  
应吉康 教授

石油大学

乐嘉锦 教授  
蒋川群 教授  
吴朝晖 教授

天津大学

李善平 教授  
骆 畔 教授  
秦小麟 教授

复旦大学

张功萱 教授

华东理工大学

华东师范大学

东华大学

上海第二工业大学

浙江大学

南京大学

南京航空航天大学

南京理工大学

南京邮电学院	朱秀昌	教授
苏州大学	龚声蓉	教授
江苏大学	宋余庆	教授
武汉大学	何炎祥	教授
华中科技大学	刘乐善	教授
中南财经政法大学	刘腾红	教授
华中师范大学	王林平	副教授
	魏开平	副教授
	叶俊民	教授
国防科技大学	赵克佳	教授
	肖 依	副教授
中南大学	陈松乔	教授
	刘卫国	教授
湖南大学	林亚平	教授
	邹北骥	教授
西安交通大学	沈钧毅	教授
	齐 勇	教授
长安大学	巨永峰	教授
西安石油学院	方 明	教授
西安邮电学院	陈莉君	副教授
哈尔滨工业大学	郭茂祖	教授
吉林大学	徐一平	教授
	毕 强	教授
长春工程学院	沙胜贤	教授
山东大学	孟祥旭	教授
	郝兴伟	教授
山东科技大学	郑永果	教授
中山大学	潘小轰	教授
厦门大学	冯少荣	教授
福州大学	林世平	副教授
云南大学	刘惟一	教授
重庆邮电学院	王国胤	教授
西南交通大学	杨 燕	副教授

# 出版说明

高等学校教材·计算机科学与技术

**改** 改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的

前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

- (1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。
- (2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。
- (3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。
- (4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。
- (5) 高等学校教材·信息管理与信息系统。
- (6) 高等学校教材·财经管理与计算机应用。

清华大学出版社经过 20 年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

# 前言

高等学校教材·计算机科学与技术

信息是人类社会和国家发展的重要战略资源。随着科学技术的快速发展,传统媒体内容正在向数字化转变。数据的交换与传输也变得更加快捷。但随之而来的日益严重的知识产权侵犯行为和基于加密的安全措施面临的严峻挑战,使得信息隐藏技术重新焕发活力。

信息隐藏是与数学、密码学、信息论、计算机视觉以及其他计算机应用技术等多学科交叉的学科,是各国研究者所关注和研究的热点。在信息隐藏研究中,可以分为基础理论研究、应用基础研究和应用研究。其中基础理论研究是建立图像信息隐藏的理论框架和若干理论模型,解决安全性度量、通信量分析等基本理论问题,以揭示信息隐藏中若干基本矛盾。信息隐藏的应用基础研究主要针对典型应用需求,研究各种信息隐藏算法和评估体系。信息隐藏的应用研究以图像信息隐藏技术的实用化为目的,研究针对各种应用的实用系统。

信息隐藏利用人类感知及数字媒体自身的冗余,将秘密信息嵌入到载体中,以对载体的使用进行跟踪,从而达到版权保护、完整性认证等目的。作为一门迅速发展的新兴交叉学科,开展其理论与关键技术的研究,具有巨大的军事和经济价值。计算机技术的发展日新月异,信息隐藏技术也不例外,还会不断涌现出新算法、新应用以及新的发展思路。

本书旨在向读者介绍这一充满活力的领域中的基本理论原理及典型算法,并介绍了三个主要分支的研究情况,以期起到抛砖引玉的作用。

本书共分 8 章。第 1 章介绍信息隐藏技术应用分类,以及数字水印及隐写术在信息安全中的地位。第 2 章介绍信息隐藏的基本原理并讨论隐写系统的分类及术语。第 3 章讨论了信息隐藏的预处理,主要涉及相关加密领域的算法及知识。第 4 章介绍隐写术的模型及数字水印模型。第 5 章分析典型的信息隐藏算法,包括位平面算法、调色板算法、空域算法及频域算法,同时还讨论融合算法。第 6 章重点介绍数字水印技术及相关典型算法与技术。第 7 章讨论隐蔽通信,主要探讨 TCP/IP 中的隐蔽通信。第 8 章介绍隐写分析技术及相应评价指标,并分析了通用原形系统的相应算法。

对于这一领域,作者的研究可能只是以管窥豹,仅见其一斑,肯定存在不足之处,希望研究这一领域的同行给予批评和指正。

书中难免会存在问题,欢迎广大读者给予批评指正。

编 者

2008 年 4 月

# 目录

高等学校教材·计算机科学与技术

第1章 简介 .....	1
1.1 引言 .....	1
1.2 隐写术概述 .....	3
1.3 数字水印概述 .....	5
1.4 隐蔽通信概述 .....	7
1.5 信息隐藏的应用 .....	8
1.6 隐写算法综述 .....	10
1.7 隐写分析概述 .....	13
1.8 信息隐藏当前研究现状与存在问题 .....	16
1.9 本章小结 .....	17
1.10 因特网资源 .....	17
1.11 复习题 .....	18
第2章 信息隐藏基本原理 .....	19
2.1 信息隐藏的基本原理与分类 .....	19
2.1.1 纯隐写术、密钥隐写术和公钥隐写术 .....	20
2.1.2 文本、音频、图像的隐写 .....	22
2.1.3 音频中的隐写 .....	24
2.2 信息隐藏的主要术语 .....	26
2.3 数字水印系统的构成与分类 .....	28
2.3.1 数字水印系统 .....	29
2.3.2 数字水印、隐写术与加密术的区别 .....	30
2.3.3 数字水印的分类 .....	30
2.3.4 数字水印的特性与术语 .....	31
2.4 本章小结 .....	32
2.5 复习题 .....	32

<b>第 3 章 信息隐藏的预处理</b>	33
3.1 加密的预处理	33
3.1.1 伪随机数发生器	33
3.1.2 RC4 流密码	35
3.2 简单的图像信息伪装技术	37
3.3 置乱	37
3.4 混沌	42
3.5 本章小结	44
3.6 复习题	44
<b>第 4 章 信息隐藏模型</b>	45
4.1 隐写术模型分析	45
4.1.1 Simmons 模型分析	45
4.1.2 通信系统模型分析	46
4.1.3 隐写术的安全模型分析	47
4.1.4 基于通信的水印模型	48
4.2 数字水印空间模型	48
4.3 感知模型	49
4.3.1 人类感知	49
4.3.2 评价的基本指标	50
4.3.3 Watson 感知模型	50
4.4 本章小结	51
4.5 复习题	51
<b>第 5 章 信息隐藏算法</b>	52
5.1 信息隐藏算法概述	52
5.2 位平面算法	53
5.2.1 位平面算法概述	53
5.2.2 位平面算法实现	53
5.2.3 嵌入算法步骤和程序	55
5.2.4 实验和实验结果分析	57
5.3 调色板算法	59
5.3.1 调色板算法原理	59
5.3.2 调色板信息隐藏算法实现	60
5.3.3 调色板信息隐藏算法容量实验	62
5.4 空域信息隐藏算法	65

5.4.1 最低有效位算法原理 .....	65
5.4.2 最低有效位算法实验 .....	66
5.4.3 Hide and Seek 隐写软件分析与实验 .....	67
5.5 频域变换信息隐藏算法 .....	70
5.5.1 离散傅里叶变换 DFT .....	70
5.5.2 离散余弦变换 DCT .....	75
5.6 小波域信息隐藏算法 .....	85
5.6.1 离散小波变换 DWT .....	85
5.6.2 小波变换实现信息隐藏 .....	91
5.7 统计算法 .....	94
5.8 图像融合算法 .....	95
5.9 本章小结 .....	97
5.10 因特网资源 .....	97
5.11 复习题 .....	97
<b>第 6 章 数字水印 .....</b>	<b>98</b>
6.1 数字水印算法概述 .....	98
6.2 空域数字水印算法 .....	99
6.2.1 最低有效位算法 .....	99
6.2.2 Patchwork 算法 .....	100
6.3 变换域算法 .....	102
6.3.1 DCT 算法 .....	103
6.3.2 DWT 算法 .....	105
6.4 可见与不可见数字水印算法 .....	107
6.5 可逆水印概述 .....	110
6.5.1 可逆数字水印现有算法 .....	111
6.5.2 基于纠错编码的差值扩展可逆数字水印 .....	112
6.6 免疫数字水印算法 .....	116
6.6.1 SRIW 形式化描述 .....	116
6.6.2 SRIW 实现方法 .....	117
6.6.3 SRIW 安全性分析及评价标准 .....	118
6.7 多重数字水印 .....	120
6.7.1 多重数字水印概述 .....	120
6.7.2 鲁棒性和脆弱性相结合的双重数字水印 .....	121
6.7.3 基于 CDMA 的多重数字水印算法 .....	124
6.8 本章小结 .....	125
6.9 复习题 .....	125

<b>第 7 章 隐蔽通信</b>	126
7.1 隐蔽通信概述	126
7.2 隐蔽通道	127
7.3 TCP 隐蔽通信	128
7.3.1 TCP 协议概述	128
7.3.2 TCP 隐蔽通信的实现	130
7.4 IGMP 中的隐蔽通信	132
7.5 IP 中的隐蔽通信	134
7.6 本章小结	137
7.7 复习题	137
<b>第 8 章 隐写分析技术</b>	138
8.1 隐写分析概述	138
8.1.1 隐写分析定义	138
8.1.2 隐写分析分类	138
8.2 隐写分析评价指标	140
8.3 隐写分析通用原型系统	141
8.4 隐写分析算法	141
8.4.1 专用隐写分析算法介绍	141
8.4.2 通用隐写分析算法介绍	142
8.4.3 GPC 隐写分析法	143
8.5 本章小结	146
8.6 复习题	146
<b>参考文献</b>	147

## 简介

### 本章目标

- 理解信息隐藏技术。
- 了解书中讨论重要主题的概况。
- 理解隐写术、数字水印和隐蔽通信。
- 理解隐写术和数字水印在信息安全中扮演的中心角色。

在“引言”之后，本章先介绍信息隐藏的三个重要分支，即隐写术、数字水印和隐蔽通信；然后简要介绍本书的每一部分。

### 1.1 引言

随着网络的应用越来越普及，信息安全成为很热门的研究领域，信息安全主要分为两大领域——加密技术与信息隐藏技术，本书主要介绍和研究信息隐藏技术。

信息隐藏是一门交叉学科，它涉及数学、密码学、信息论、计算机视觉以及其他计算机应用技术，是各国研究者所关注和研究的热点。其原理是利用载体中存在的冗余信息来隐藏秘密对象，以实现保密通信或者实现数字签名和认证。信息隐藏与信息加密是不尽相同的，信息加密是隐藏信息的内容，而信息隐藏是隐藏信息的存在性，信息隐藏比信息加密更为安全，因为它不容易引起攻击者的注意。但两者又不能截然分开。信息隐藏打破了传统密码学的思维范畴，从一个全新的视角审视信息安全。与传统的加密相比，信息隐藏的隐蔽性更强，在信息隐藏中，可以把这两项技术结合起来，先将秘密信息进行加密预处理，然后再进行信息隐藏，则秘密信息的保密性和不可觉察性的效果更佳。

信息隐藏技术的推动力有两个方面：第一方面是需要保护知识产权的用户。目前通过互联网，信息能被轻易地传递和复制，这使信息的知识产权变得更难保护。数字水印技术的使用提供了在文档或图像中插入版权提示，用于保护信息的知识产权。数字水印经常是小的图像或文本，在整个文档或图像中不断地重复。相似的技术是嵌入数字指纹和系列号。指纹的优势在于它能用于追踪对源文件的复制以及可以作为起诉的有力工具。

第二方面是对隐藏信息有兴趣的人们，希望以秘密的方式传送信息并且避免第三方接收者的察觉。在这种情况下，隐藏的信息比用来运送它的载体更重要。隐写术经常与加密

术一起用于限制未授权的信息访问。加密术是指通过加密或者以打乱信息的方式来使信息只能到达指定接受者并解密信息。当发送加密的信息时就明显地表明,已经发生了某种形式的通信,并发送了加密的消息,使消息不能被非指定的对象解读。隐写术经常用来隐藏消息的存在。

在信息隐藏中,目前广泛使用的是数字水印技术、隐写术和隐蔽通信。数字水印和隐写术是信息隐藏的两个重要分支。在 20 世纪 90 年代早期,与加密技术相比,信息隐藏技术并没有引起学术界的更多关注。但是随着计算机和网络通信技术的发展与普及,数字化的音像制品和其他电子出版物的传播和交易变得越来越便捷,未授权的复制和侵权盗版行为日益严重。在这种大背景下,信息隐藏这一古老的技术重新焕发了活力。研究者首先想到的就是在数字产品中藏入版权信息和产品序列号以防止侵权行为。随着研究的进一步深入,目前信息隐藏技术越来越受到各届的关注,主要是因为版权的拥有者想保护其版权不被盗版,所以信息隐藏中的数字水印技术得到了空前的发展,目前广泛使用于音乐、电影、书籍和软件的防盗版中。

信息隐藏具备的特性如下。

- 不可感知性(imperceptibility) 有时也称为隐蔽性。这一特性是信息隐藏最必要的条件。载入信息的伪装载体与原载体(没有嵌入秘密信息的载体)应当大体上是很接近的,从人的视觉上应该感觉不到任何变化。传统的信息隐藏是将秘密信息嵌入到一般信息中,使得人只看到一般信息,而看不到秘密信息。在不改变原有信息内容的前提下,使一般信息与秘密信息的总体容量远远超过一般信息的容量,这样,传输速度会减慢,也会使人生疑,从而使秘密信息被截获的几率加大。所以,对于信息隐藏而言,重要的是,载体在加载秘密信息前后的大小一般不应变化很大。
- 不可检测性(undetectable) 不可检测性是信息隐藏的目的,如果检测到信息隐藏的存在,说明信息隐藏本身已经失败。
- 容量(capacity) 在保证不可感知性和不可检测性的前提之下,希望载体能嵌入的数据容量越大越好,但容量增大,会降低不可感知性和不可检测性,所以要均衡这三种特性。秘密信息容量越大,隐藏的难度系数越大;图片要比文本更难隐藏;秘密信息与载体信息越接近,保密的效果就会越好。
- 鲁棒性(robustness) 是指嵌入水印后的数据经过各种处理操作和攻击操作以后,不导致其中的水印信息丢失或被破坏的能力。攻击操作一般包括模糊、几何变形、放缩、压缩格式变换、剪切等。
- 安全性(security) 指水印不易被复制、伪造、非法检测和移去,文件格式的变换不会导致水印丢失。
- 复杂性(complication) 指水印的嵌入和提取算法复杂度低,便于推广应用。

在互联网开放的环境中,正在广泛使用着各种信息隐藏工具,信息隐藏工具是一把双刃剑,既可以保护信息的安全,也可为恐怖分子所利用,所以研究信息隐藏有很重要的意义。

下面分别概括性地介绍信息隐藏中的三个重要领域:隐写术、数字水印和隐蔽通信。

## 1.2 隐写术概述

隐写术(steganography)来自于希腊词根  $\sigma\tau\epsilon\gamma\alpha\nu\zeta, \gamma\rho\alpha\varphi\epsilon\nu$ , 含义是隐写。它的起源可以追溯到公元前 440 年。隐写术有史可考的第一个记录是希腊史学家 Herodotus 的叙述。在由 Herodotus 写的历史记录中, 给出了两个隐写术的例子。第一个是 Demeratus 的例子。在波斯的一个希腊人, 为了通知即将到来的入侵, 他在木板上写上信息并用蜡涂在木板上, 信息不见了, 信使成功地将空白的木板带到了斯巴达。第二个是 Histiaeus 的例子, 他剪去了他最信任的奴隶的头发, 然后将信息刺到它的头上。当这个奴隶的头发长出来后, 再派他发送这些隐藏的信息。

隐写术的最普通形式是使用看不见的墨水来写消息。在第二次世界大战期间, 有很多联军使用这种方法。这些消息经常使用果汁、牛奶或者尿来写, 当加热这些消息的载体时, 将变黑显示消息。当隐写墨水技术已经很容易地被破解时, 人们开始使用 null ciphers, 这个词是指未加密的信息, 对第三方而言, 是很难觉察的。如下面的一个例子:

Fishing freshwater bends and saltwater coasts rewards anyone feeling stressed.  
Resourceful anglers usually find masterful leapers fun and admit swordfish rank overwhelming anyday.

取每个词中的第三个字母, 就得出了其中隐藏的消息:

Send Lawyers, Guns, and Money.

在第二次世界大战中德军发明了微点技术(microdots), 该技术将重要的情报缩小数十倍, 伪装成任何印刷品的字母或标点符号, 有效地传递大量情报。美国联邦调查局局长胡佛(FBI Director J. E. Hoover)还曾夸赞德国人的发明真是间谍活动的一大杰作。微点技术处理过的情报, 需要在接收方使用显微镜能阅读这些情报, 对于非知情者而言, 通常根本觉察不到通过微点技术处理信息的存在。

隐写术提供了缩微拍摄的可能性, 它可以在衣服或行李的间隔中私带秘密。在西班牙对普鲁士的战争中很流行。在 21 世纪, 政府已经开始使用隐写术来保护真钞, 来防止假钞。它们使用特殊的油墨、染料、嵌入线和微波等来鉴别钞票的真伪。随着 Internet 技术的成长, 隐写术也继续发展。

目前隐写术最常见的用法是将秘密信息隐藏到另一个载体中, 载体可以是图像、音频、视频和文本或者其他二进制数字编码。隐藏的信息可以是纯文本、密码图像或者其他比特流。网络中使用的大部分文件格式是.bmp,.doc,.gif,.jpeg,.mp3,.txt 和.wav 等。载体和隐藏的信息生成了伪装载体。隐写密钥可以进一步保证隐藏信息的安全性。隐写处理的过程可以概括如下:

cover medium + hidden information + stego - key = stego - medium

其中掩密密钥(stego-key)可以用于隐藏和对信息解码。隐写术需要特定的软件。使用隐写术的目标是: 在传输隐藏信息时避免引起注意。如果引起注意, 则隐写失败。

在人类视觉上并不能感知到隐写术处理后的图像质量的下降, 因此在互联网上的任何

图像都可以隐藏信息，并且不被怀疑。在美国 USA Today 杂志上的一篇文章写道：恐怖组织使用隐写术进行信息交流而没有被发现。根据美国专家分析，这篇文章缺乏技术信息来支持这个论点。但是在网络快速发展的今天，人们无时无刻不在使用网络进行信息的共享和交流，安全的通信环境是人人所需要的，而使用隐写术正好能够完成个人的安全私密通信。

在数字世界中，隐写术和加密术都是保护信息不被未授权的第三方看到，都是保护信息的很好手段。但是这两种技术都不是无懈可击的，都可能被破解。这就是专家建议使用这两种技术来保护信息的原因所在。此外，隐写术还经常使用在很重要的领域。如在某种情况下，不能自由通信，甚至是在受监控的情况下，为了保护秘密通信而又不想使用加密来引起怀疑的情况下，使用隐写术是一种很好的选择。

隐写术是未来的互联网安全中很重要的一部分，也是在开放的环境中如何保护私密性的关键技术。隐写研究的推动力基本在于自身加密系统的局限性以及需要在开放环境中完全的私密性。经过隐写术处理过的文件，一般察觉不到隐蔽信息的存在，这样只有接收方才能知道隐藏信息的存在并能提取这些秘密消息。可以说，隐写术完全满足了人们对私密通信的需求。

下面给出将密文放入一幅.bmp 文件中的例子。密文是密钥，密钥是经过加密算法生成的，需要经过 Internet 这种开放的环境传送出去。通过编写的隐写软件，将密钥嵌入了原始图片。如图 1.1 和图 1.2 所示，对读者而言，这两幅图是相同的，但在第二幅图中已经嵌入了密钥，所以对第三方而言，是感觉不到密钥的存在，这样就达到了秘密传输密钥的目的。当接收者接收到图像，可以用隐写软件提取出密钥。

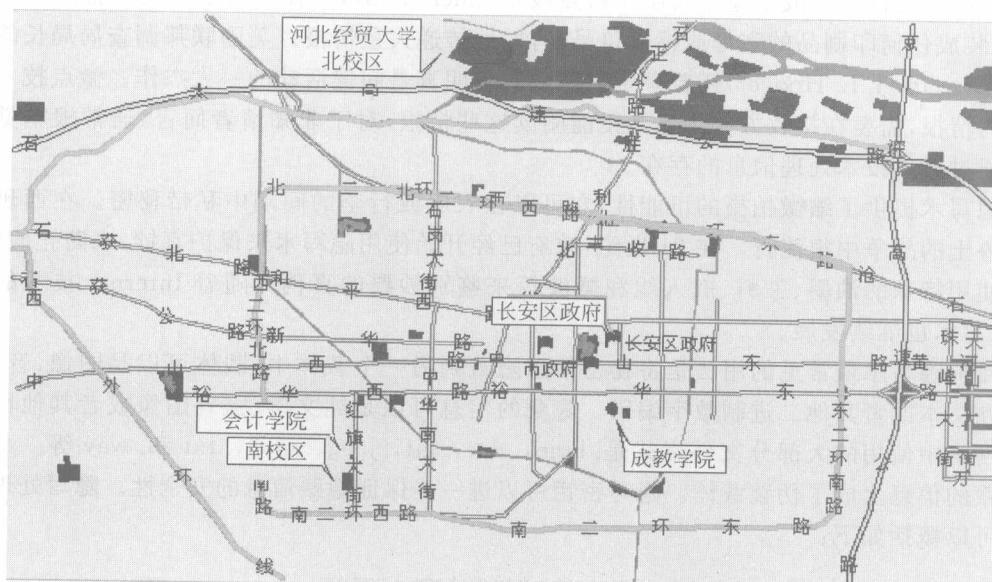


图 1.1 原始图像

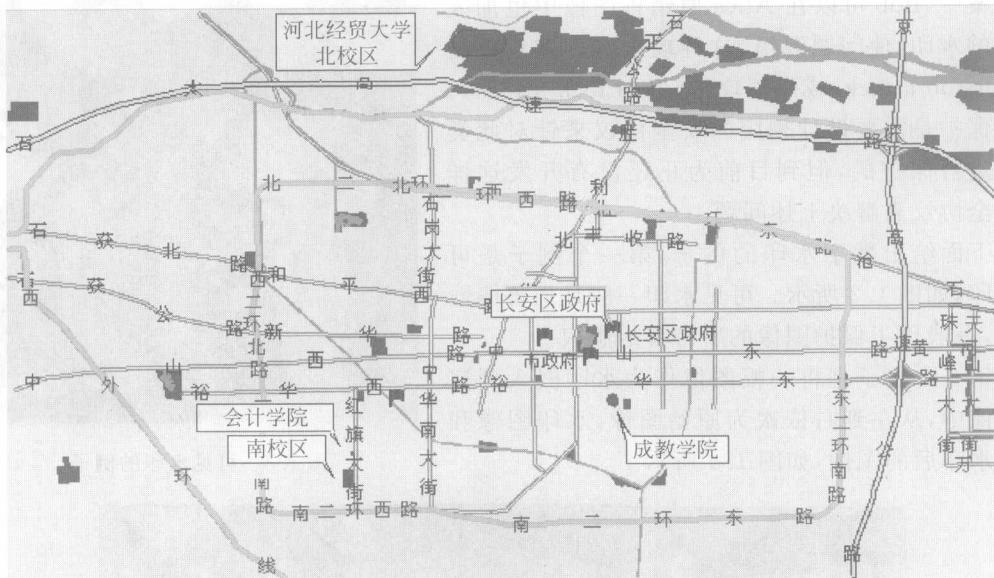


图 1.2 嵌入密钥的图像

### 1.3 数字水印概述

水印可以追溯到德语词汇“wassermarke”，纸水印在 1282 年出现在意大利，这些最早的水印是通过纸模中加细线模板制造出来的，在有细线的区域，纸更透明、更薄。早期的功能是识别纸的制造商。到 18 世纪，欧洲和美国制造的产品中，纸水印用于纸币和其他文件的防伪。对于数字水印(digital watermarking)，不同的人有不同的理解，但一般而言数字水印就是不可觉察地将秘密信息嵌入载体信号来传送秘密数据。将信息嵌入到其他对象/信号的过程称为嵌入水印。数字水印经常用于版权保护和拷贝保护，其主要应用是图像/视频的版权保护，拷贝保护是指限制或禁止未授权的保护。拷贝保护的最好例子是加密的数字 TV 广播，通过使用许可服务器和访问控制来保护软件的合法使用。版权保护是将版权信息插入到数字对象而对数字对象的质量没有任何损害。当产生数字对象的版权纠纷时，可以从数字对象中提取嵌入信息来证明数字对象的所有者。它可以广泛用于未授权复本的追踪。关于水印最原始的论文是在 13 世纪。由于许多摄影师并不十分信任非可见水印，所以目前的可见水印都是将自己独特的标识直接嵌入到载体之上。在 17 世纪，法国 Claude Lorrain 引入了水印方法来保护自己的版权，在 1710 年英国引入了版权法。

可以通过一个例子来说明如何进行版权保护。Alice 是版权的所有者，她将自己的水印信息嵌入了载体对象，锁定了原载体并开始销售带有水印的图像。Bob 试图将自己的水印嵌入到 Alice 处理过的伪装载体。然后锁定再次嵌入水印的图像并进行销售。为了证明图像的所有者，Alice 和 Bob 都能提取相应的水印来证明自己是拥有者。从前面的叙述中可知，Alice 的销售图像中只拥有自己的水印，而 Bob 销售的图像中还包含 Alice 的水印。这是否就可以表明，Bob 不是销售图像的所有者。但是，情况并非如此简单，在各种不同的水

印方案中,Bob 可以在 Alice 原销售图像中也加入自己的水印,使问题混淆,这种攻击称为倒置攻击(inversion attack)或死锁攻击(dead lock attack)。版权保护是需要设计相应的安全协议来针对此类情况进行保护的,但到目前为止还没有开发这样的安全协议来解决上述问题。

下面给出数字水印的例子,第一个例子是可见水印,如图 1.3 所示。可见水印一般是强鲁棒性水印,一般用于保护图像的所有者的版权。

第二个例子是将一幅图像作为水印嵌入到宿主图像中,从左到右依次为原始图像,水印图像和嵌入水印后的图像,如图 1.4 所示。

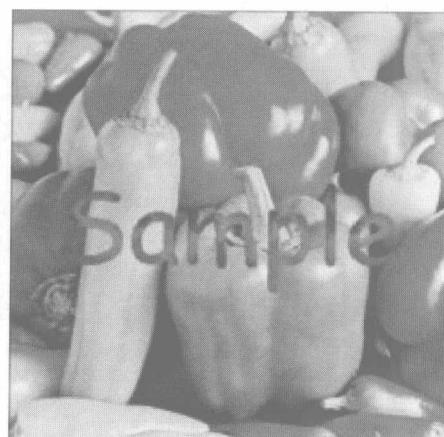


图 1.3 可见水印的例子

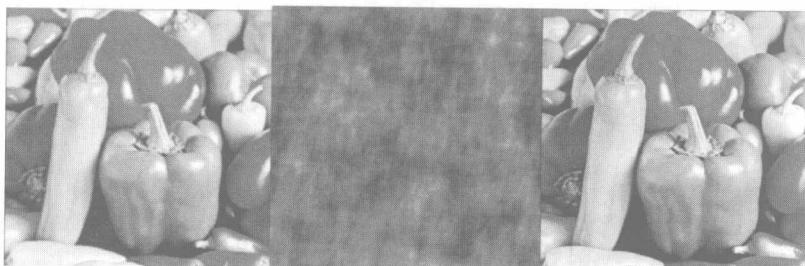


图 1.4 原始图像、水印图像和嵌入水印后的图像

现在,数字水印软件既有商品化产品,也有供研究用的免费软件。商品化数字水印软件的厂商主要包括:

(1) Digimarc 公司(<http://www.digimarc.com>)。美国 Digimarc 公司成立于 1995 年,是最早从事数字水印软件开发的企业之一,其产品主要面向多媒体版权保护、认证和电子商务等领域。

(2) Signum 技术公司([http://www.signumtech.com/index\\_ns.html](http://www.signumtech.com/index_ns.html))。这家英国公司成立于 1997 年,所开发的 SureSign 系列数字水印产品主要面向数字摄影、多媒体、网络发行、电子商务和医学影像等领域。

(3) Aliroo 有限公司(<http://www.aliroo.com>)。该公司成立于 1993 年 12 月,主要开发各种基于密码学的网络安全产品和数字水印软件。Aliroo 公司与 Digimarc 公司达成了系列技术协议,其开发的数字水印软件 ScarLet 可以直接使用 Digimarc 公司的认证服务。

(4) Alpha 技术公司(<http://www.generation.net/~pitas/>)。Alpha 公司是专门从事计算机图形学、图像处理、计算机视觉等专业软件开发的企业,其开发的数字水印产品 EIKONAmark 在技术上有很多特色,非常适于数字图像的版权保护。

(5) MediaSec 技术公司(<http://www.mediasec.com>)。该公司是一家专业的信息隐藏技术公司,其开发的 SysCop 系列产品主要面向数字水印、隐蔽标识和隐蔽通信。SysCop 系列产品最突出的特点是允许在图像(PPM/PGM/PBM、GIF、TIFF 和 JPEG 格式)和视频信号(MPEGI 和 MPEGII)中灵活地隐藏各种长度的信息。

供研究用的免费软件有：

(1) S-Tools(<ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/stools4.zip>)。是空域数字水印软件,支持.wav格式的音频文件和.gif、.bmp等格式的数字图像文件。S-Tools处理24位真彩色图像的速度很快,对于索引色图像,根据用户的选择,可以还原成真彩色图像或通过削减颜色数量添加水印。

(2) Hide and Seek(<ftp://ftp.csua.berkeley.edu/pub/cypherpunk/steganography/hdsk41b.zip>)。是空域数字水印软件,它对图像的限制较多,只能处理256色图像,图像尺寸被限制为320×320、320×400、320×480、640×400、1024×768。

(3) Hide4PGP(<http://www.rugeley.demon.co.uk/security/hide4pgp.zip>)。是一个典型的使用LSB算法的数字水印软件,用于在8位或24位BMP图像中嵌入水印。对于24位真彩色图像,可选的隐藏位数为1、2、4、8几种。对于8位索引色图像,Hide4PGP引入的噪声很明显。

(4) StegDOS(<ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/stegodos.zip>)。是早期的运行在DOS下的水印软件,使用的也是LSB方法,效果比较差。

(5) White Noise Storm(<ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/wns210.zip>)。是典型的基于扩展频谱技术的数字水印软件,隐藏效果非常好,但数据量偏小。

(6) Mandelsteg(<ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/>)。是一个提供源代码的空域数字水印软件。

(7) Jsteg-Jpeg(<ftp://ftp.funet.fi/pub/crypt/steganography>)。Jsteg-Jpeg是专门针对JPEG图像格式开发的数字水印软件,水印隐藏在DCT变换域上。从处理后的图像上很难看出隐藏数据的痕迹,但对比添加水印前后的DCT谱,可以发现嵌入水印后图像的DCT变换系数有明显的阶梯效应。

(8) UnZign(<http://altern.org/watermark/>)。UnZign是早期的(1997年)数字水印测试工具。

(9) StirMark([http://www.cl.cam.ac.uk/~fapp2/watermarking/image\\_watermarking/stirmark](http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark))。StirMark是一个在数字水印研究领域中非常有名的测试工具,由剑桥大学开发,其版本更新速度很快。StirMark可以从多方面测试水印算法的鲁棒性,用于测试的攻击手段包括线性滤波、非线性滤波、剪切/拼接攻击、同步性破坏攻击等。许多公开发表的数字水印方面的论文都以StirMark的攻击结果作为衡量水印算法好坏的标准。

## 1.4 隐蔽通信概述

信息可以隐藏在什么地方?可以隐藏在Internet的大多数地方。除了Web页面上的文字、图像和音频等多媒体介质之外,还可将信息隐藏在Internet上使用的各类协议中,如网络互联必用的标准协议组TCP/IP,就经常允许使用标志和特殊的保留字段在两台计算机之间进行信息传输。如果使用恰当的工具,将信息能插入到这些字段,那么就可以传输大量秘密信息而不被察觉。这种技术的优势是协议头很少被别人阅读,所以这正是隐藏信息的最理想的地方。这种方法的不利之处在于设定的防火墙能过滤掉数据包,如果包的保留

字段中包含不合适的信息,那么就会因过滤而丢失数据包。

信息隐藏处理后的数据可以通过开放的信道进行传输,它可以通过网络中的防火墙和任何网络入侵系统的检测。但还有另外一种可能,那就是实际上大部分入侵检测系统都能检测到负载中的隐藏数据,只是提取存在一定问题,所以为了使信息隐藏的数据不公开暴露在公共信道中,人们开始通过隐蔽通道来进行秘密通信。通过协议建立秘密通道,允许未授权访问来通过授权的防火墙。一些工具可以使传输有价值的数据流量看起来像正常的网络流量。这样的工具有 loki。它可以将数据隐藏到 ICMP 流量中。

使用隐蔽信道的思想是维护信息系统的安全性、完整性和一致性的人员所不能接受的。1985 年美国国防部公布的“对计算机系统安全性评估”一文中对隐蔽信道描述如下:“any communication channel that can be exploited by a process to transfer information in a manner that violates the system’s security policy”。在实际中隐蔽信道有两种类型:存储信道与分时信道。

隐蔽存储信道包括所有的设备可以直接或间接地通过处理器写到相应的存储位置或者由其他处理器以直接或间接的方式读取。隐蔽分时信道包括所有设备允许一方处理器给另外一方发送信息,通过调制它自己所使用的系统资源,以这种方式改变所使用的预留响应时间,这样对方的进程就可以向这个进程提供消息。

本书第 7 章将专门介绍隐蔽通信。

## 1.5 信息隐藏的应用

信息隐藏是将字符、语音、图像隐藏到数字数据中,并使它们看起来像别的字符、声音和图像,把秘密转换成无害的噪声以便使其秘密通过网络,并在位流的“海洋”之中消失。信息隐藏方法可分成三种截然不同的模式:插入(injection)、代替(substitution)和繁殖(propagation)或者称为“产生一个新文件”。在第一种和第二种模式中,经常使用隐写密钥,用密钥提供隐藏和提取的过程,来防止或者拒绝未授权用户的访问。

插入隐写能像被用户期望的方式工作,在这种方式中,负载或者要嵌入的数据要放到载体文件(包括文本、图像、音频或程序文件)中。这样做将增加主机文件的容量,为了防止终端处理或者应用(如字处理程序、图片浏览器和媒体播放器等)将嵌入在载体中的数据揭示出来,大多数的文件类型不适合使用这种方式,原因就在文件受到了处理过程或者使用隐写方法的影响而改变了容量。

代替隐写法是将需要隐藏的信息替换了那些在载体文件中不重要的部分,但是载体文件必须通过相应应用才能正确地显示出来。可执行的载体文件可代替的部分可以是程序模块或者可执行的代码部分,这些模块和代码很少使用或几乎不使用。这种方法有时称为“bit-twiddling”或者“bit-tweaking”(“捻弄一点儿”或者“拧一点儿”),这种方法可能使文件质量下降,例如在录像或者静止图像时出现偏差,在音频文件中听得见噪音,或者可执行文件执行时出现处理错误。

繁殖隐写经常使用引擎产生器(generation engine),当有负载时产生输出文件,通常文件的内容被称为模拟(mimic),而每次生成的文件可以表示为任意格式的图片、音频文件或其他格式的文件。在网络中,许多恶意代码就是通过插入方式进入程序,然后通过自身繁殖

来破坏源程序的。

信息隐藏技术的应用很广泛,大致可以归类如下。

- 秘密通信 秘密通信隐藏了通信双方以及通信过程的存在。
- 版权保护 授权的水印以不可感知的方式嵌入到多媒体中。
- 认证和篡改检测 可以对数字作品进行认证,并且对篡改进行检测。
- 盗版追踪 用来跟踪作者或购买者的多媒体的某种备份。
- 信息标注 信息标注是指隐藏一些信息于载体介质中,用于解释与介质有关的一些内容。
- 复制控制与访问控制 将数字水印嵌入来表示某种复制控制和访问控制限制。
- 信息监控 需要对某些信息进行控制,可以使用信息隐藏技术。
- 票据防伪 票据防伪是保证票据中隐藏的水印信息在打印之后仍然存在,可以保证票据的真实性。
- 军事和其他一些情报机构,需要秘密的通信手段 在现代战场上对这些敏感信号的检测可能导致对发报员的快速攻击,军方通信中往往采用发散谱调制或大气散射等传递技术,来保证信号的准确传送。
- 恐怖分子也在研究使用信息隐藏技术 通过对资料的研究,美国反恐组织分析,在“9·11”事件中,恐怖分子使用的就是隐写术,将指令隐写到多媒体中(如图片),在互联网上传送,对于隐写处理过的图片,没有专门的隐写分析工具,很难发现。
- 电子商务的兴起,使信息安全的呼声更高,除了加密术之外,人们更关注信息隐藏中的认证技术。

那么上述的应用中,信息隐藏是如何实现的?在数字时代将如何隐藏信息呢?众所周知,计算机是基于二进制的,由0和1表示文本和图形。美国标准信息交换代码ASCII码表示文本和某些特殊字符。ASCII码使用一个奇偶位和7个数据位表示在英语中的一个字符,例如大写字母A由1000001表示。

还有把信息和噪声放到一幅图片或是一组声音里,数字文件是由一组用来表征光线和声音的强度在时间和空间上精确点的数字组成。例如,一张图片上的一个点,可以由220个蓝色单位在总阵列中从0~255区间内的变化所表征,当这个点由220个蓝色单位转换成219个蓝色单位时,用肉眼不可能察觉到它的变化。

数字图像由基本元素像素组成,每个像素信息的强度与三基色有关:红色、绿色和蓝色。信息可以存储在单字节(8位)或三字节(24位)中。例如在8位图像中白色用11111111的值表示,黑色用00000000表示。当前的信息隐藏技术依靠使用掩护对象(图像、文档或声音文件等)作为载体。通过伪装工具(stego-tool)将秘密信息分离成单独的位,然后嵌入到掩体对象中。许多工具都使用密码或口令,作为掩密-密钥(stego-key),这在提取隐藏信息时是非常必要的,整个过程包装出的文件称为伪装对象。

很流行的信息隐藏技术是使用文档中的额外空间。这些空间可以包含特殊字符。对于信息隐藏这是一种很简单的技术,经常很容易被检测到,从而导致信息隐藏失败。在文字处理器中打开这类文件时,与众不同的空间变得显而易见。将文档重定格式就能移除这些信息。音频文件也是用于信息隐藏的很好载体。因为声音文件有很大的存储空间并且不会引起注意。特别是MP3Stego工具,这个工具很有用,能用于隐藏信息并且维持CD声音的

质量。

目前最流行的伪装对象是数字图像,因为它们有效的载荷(隐藏的信息)很大。如果图像为 $640\times 480$ 像素和256色(8位)几乎能隐藏大约300KB的信息。在高质量的图像中,是 $1024\times 768$ 像素和24位,能隐藏大约2.3MB的数据值。由于这些文件都要使用压缩技术来压缩图像的大小以便于在Internet上传输。目前有很多可用的压缩算法,最常用的有三种图像格式:BMP、GIF和JPEG。在Internet上选择用于隐写术的图像时,我们首先选择BMP和GIF。因为这两种格式的图像能提供无损压缩。压缩的图像可以精确地还原。JPEG压缩算法使用移动点计算传送的图像到整数隐列。这种变换处理能产生错误,它能消除一部分图像,但人眼不能觉察,这对于隐藏的信息可能造成破坏或改变。在图像中嵌入数据可以使用两种技术:空间域工具(image domain tools)和交换域工具。空间域工具也称为bit wise methods,是对图像的最低有效位进行处理。在这种方式下每个像素左边的最低位被秘密信息位所代替。因为LSB只能包含1或者0,所以在嵌入秘密消息时接近一半的比特位不需要发生改变。对于低质量的8位彩色图像使用LSB将引起显著的颜色变化。所以高质量的图像才适合用做载体。但这中间也有一个例外,那就是灰度级图像。在灰度级图像中使用8位定义256个在黑白之间的灰度级。因此使用LSB时就像是生成一个新的图像,以前在调色板中的灰度级均不起作用。一些在空间域中很流行的工具有Hide and Seek、Mandelsteg、Steganos、StegoDos、S-TOOLS和White Noise Storm。

交换域工具利用如离散余弦变换(discrete cosine transformation,DCT)或者小波变换(wavelet transformation)等算法在图像不重要的位置来隐藏信息。使用交换域算法的隐写工具生成的伪装图像具有更高的鲁棒性、更能有效地抵御如压缩、剪裁等各种形式的攻击以及图像处理。对于JPEG格式的图像,在交换域中的隐写工具有:Jpeg-Jsteg、JPHide、Outguess、PictureMarc和SysCop。JPEG图像格式使用DCT技术将每个连续传输的 $8\times 8$ 的像素块转换成64DCT系数数,DCT系数的量化系数的最低有效位作为秘密信息嵌入的冗余位。对于其他格式的图像文件,如GIF格式的文件,图像的可视的结构中存在一定程度的位层。隐写系统对最低有效位的改变通常会引起图像视觉的变化。而这点JPEG图像是不存在的。单纯地对DCT系统的改变将影响所有的64位像素。所以对JPEG格式的图像而言,不存在视觉攻击。交换域中可使用的工具包括管理算法和图像传输,如DCT。DCT技术用于压缩图像,如JPEG、MJPEG和MPEG,在进一步处理过程中,每个像素值都转化成频率值,对于这类的隐写分析便存在一定的困难。另外的方法是改变图像的属性:如亮度或者颜色调色板。这种方法隐藏的信息容量更大。可以达到载体的30%左右。JPEG图像广泛用于因特网上,是因为它们在压缩之后,图像的质量并不降低。

本书将在相应章节中详细地介绍信息隐藏在不同载体中的应用。

## 1.6 隐写算法综述

在信息隐藏算法中,主要有空间域算法和变换域算法。最典型的空间域信息隐藏算法为LSB算法。在变换域算法中,正交变换的形式可以有离散傅里叶变换(DFT),离散余弦变换(DCT)和小波变换(WT)等。由于变换域算法利用了人眼对于不同空间频率的敏感性,在适当的位置嵌入信息具有更好的鲁棒性和不可觉察性,并且隐藏信息的容量也较高,

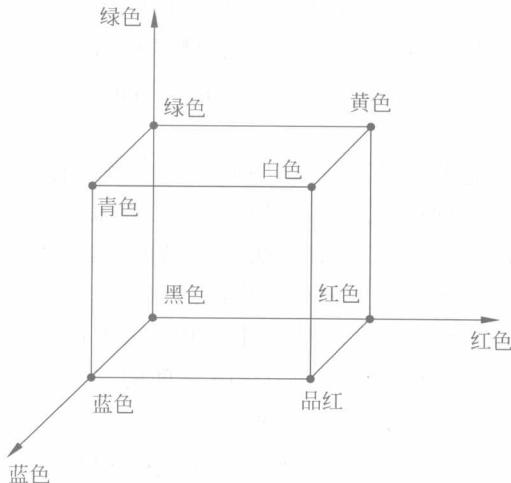
所以变换域隐藏算法比空间域算法复杂。

下面我们具体分析常见的方法：

- 最低有效位插入法(least significant bit insertion)
- 伪装与过滤(masking and filtering)
- 算法和变换(algorithms and transformations)

这些技术中的每种算法都获得了不同程度的成功。

这里将以彩色图像作为例子,因为大多数可用的隐写软件都将彩色图像作为载体对象。另外,图像也都不限于彩色图像,但是特定的彩色像素提供了额外的冗余空间。在计算机中的RGB模式就是按照自然界里三种基本色混合的原理而做的一种模式,就是red(红)、green(绿)、blue(蓝)混合,通过三种基本颜色亮度值从0~255不同组合中产生出其他各种颜色,这种模式叫加色模式。通常使用的电视屏幕和电脑屏幕上的显示就是这样的模式,在没有图像时,屏幕是黑的,若R、G、B三色亮度都为255时混合叠加打在屏幕上时则显示成白色。就是加起来是白色的意思,叫加色模式。而CMYK模式是一种印刷模式,是用cyan(青色)、magenta(品红)、yellow(黄)和black(黑)四种颜色混合,其实就是四种颜色的油墨混合。这种模式叫减色模式。因为印刷是印在纸上的,纸一般都是白色的,有油墨涂上去才显示颜色,当C、M、Y三种颜色油墨完全混合则产生了黑色,因此,减去颜色才是白色,所以叫减色模式。模式图如图1.5所示。



目前的关键创新在于使用那些包含了许多随机信息的白噪声。秘密信息可以替代这些白噪声,如果处理得很好,这些秘密信息也会像随机的噪声一样。最流行的方案就是使用数字图像,所以我们需要更进一步地研究这项技术。在数字图像或音频、视频中存在着许多白噪声,可以把白噪声理解成无声中的嘶嘶声或录像带中的空白部分。数字图像是以像素的阵列来存储的,在彩色方案中,每个像素用8位表示,这样就可以提供256种选择,而色彩元素用RGB,三种颜色在数量上是不均等的。即 $256/3=85.3333$ 。这里我们将使用1位来分析彩色频谱。

可见光频谱是红到紫的连续集,所以对于红色将有255种不同的红色,对于绿和蓝也是

如此。所以对于每个独立的像素,将有 16777215 种表示方式。作为对比,对于偏移打印的空间能提供大约 4000 种颜色,而图像可能包含大约 6000000 种颜色,而人眼大约能分辨 10000000 种颜色,这也是 CRT 使用 64 位色彩的原因。十进制数中三的变化等同于两位最低有效位的变化,即从 1-1 到 0-0。相反地会增加两个 LSB 元素的值从 0-0 到 1-1。所以从数学角度可以看出,如果给定自由的从 0 到任何顺序的范围或 1 到任何顺序的范围,每个像素的独特的颜色元素可以通过十进制数 3 的变化来体现。所以对于给定的像素,将有  $3 \times 3 \times 3$ ,即 27 种可能的比特值,所以有 26 种可能变化的颜色值。考虑到人眼对彩色像素的感知,当三种颜色位同时变化时,或增大或减小,将没有任何颜色的变化,变化的只是颜色的亮度。人眼是否能注意这一变化呢?如果图像是由计算机产生并且使用放大技术通过源图像与隐写图像进行对比,而且图像是在白色暴风雪中的北极熊、或者在纯蓝色天空下粉红色的风筝这样的图片,人的视觉可能会感觉到变化,所以载体图像是需要进行选择的,应该使用最大的可用的色阶 32 位或 8 位(灰度图像)。不要使用 4 位最低有效位,即使并不明显隐藏的秘密信息也会显现的。那么我们如何隐藏我们的信息呢?从根本上讲,要考虑两个因素:分辨率和图片的大小。

数字图像中的每个像素用 8 位表示,将有 128 种表示方式,所以会有 128 个最低有效位。如果一个图像最低 4 位有效位改变时,将不影响图像的质量。这样就提供了许多可用的空间来嵌入隐蔽信息。最多 16 位颜色的值。文本每个字符是 8 位,因此在每个像素中可以隐藏 1.5 像素在载体图像中。对于  $640 \times 480$  像素的图像,能嵌入超过 400000 个字符,所以大小适中的图像可以置放于整篇文章中。将信息隐藏入载体图像的处理应为:使像素依次排列,提取每个像素值的 4 个最低有效位,然后将秘密消息替代这些值。从人眼感觉上,图像没有发生任何变化。

但是,对于追踪隐藏信息的人员而言,这些嵌入的秘密信息数据很显眼,因为它们不是随机的,可以通过统计分析软件很容易地通过排序查找出来。所以优秀的隐写术一定是使秘密信息也像白噪声一样随机嵌入。一种做法是在隐藏信息之间对其进行编码,使用好的编码,可以使嵌入的秘密信息和图片中的要替代的随机信息最相似。另一种方法是将嵌入的秘密信息随机地嵌入到载体图像中。假随机数生成器生成开始数据的值,称为种子数。然后生成一串很像随机数的数字。例如,将 0~16 之间的数作为种子数,然后将这些种子数乘以 3,再加 1。除了 8 之外,可以得到如下的序列:1,4,13,6,2,7,5,16,15,12,3,10,14,9,11,0,1,4 等,这将与随机数很像。为了将信息随机嵌入载体图像,可以将假随机序列的数字作为像素的值。如果想提取信息,需要知道假随机数生成器的种子数和算法才能提取秘密信息。

使用这种技术,将白噪声与随机的秘密信息分开是比较难的,甚至即使怀疑有隐藏的信息,但证明其存在也存在一定的难度。为了证明或破解这些随机序列,需要猜测随机种子。这种方案使当代隐写术有了似是而非的否认本领。

除了 LSB 算法之外,还有伪装与过滤。伪装和过滤技术用于信息隐藏通常针对的是 24 位图像和灰度级图像,数据水印包括版权、拥有者或者许可证等信息。而目前的信息隐藏技术拓展了这个范围。伪装技术是将秘密信息完整地隐藏到伪装载体中而不是单纯地将秘密信息放入到噪声级别中。伪装技术对秘密信息增加了冗余,这使伪装技术比 LSB 算法更适用于无损压缩图像 JPEG 格式。它更能有效地保护秘密信息不受剪切和旋转等处理的

损坏。

另外一种隐写技术是将秘密信息嵌入到压缩算法中。这种算法将数据位隐藏到最低有效的系数中。JPEG 格式图像的关键优势在于它的无损压缩方法。它使图像即使存储在相对很小的文件中时,还能保证图像的高质量。压缩数据作为整数存储,但是量子化处理需要全面的浮点运算。差错引入全面定义 JPEG 压缩方法的无损压缩特性,JPEG 图像使用离散余弦变换技术,在 JPEG 文件中,图像是由 DCT 系数组成,当一个文件在嵌入 JPEG 文件时,这些系数的相对关系改变了,与 LSB 隐写算法中的图像中确切位的改变不同,DCT 方法中是图像的系数的相对关系发生了改变。另外对于 DCT,图像也可以使用快速傅里叶转换(fast Fourier transform,FFT)进行处理,FFT 根据图像的组成频率生成无穷序列的数据点,它同样也解决了从频率数据重构信号中的旋转等问题。

小波变换是基于局部频率的基本函数。小波压缩方法能更好地表现瞬时性,就如在夜晚的天空流星划过的图像,这就意味着一些瞬时的数据信号元素也可以通过更小数量的数据进行表示,也意味着更广泛的离散余弦的传输正被使用。小波压缩更适用于对瞬时性信号特性的表示,而不适用于平滑周期性的信号。

许多交换域方法并不依赖于图像格式,所以当在有损和无损之间进行转换时,隐藏的信息是保留的。在数据隐藏时,载体图像使用 DCT 或者小波变换,并且发现有些系数低于特定的阈值,用要隐藏的信息位代替这些信息位,例如,使用 LSB 插入方法,然后进行逆向转换并将转换后的结果保存作为固定图像。从隐藏了秘密信息的图像中提取秘密信息时,找到低于特定阈值的系数,从这些系数中提取数据位,并将这些数据位连在一起成为实际的信息。

这里只是概要地介绍了信息隐藏算法,本书将在相应章节中详细讨论相应的典型算法。

## 1.7 隐写分析概述

隐写分析是相对较新的研究领域,隐写分析就是“the process of detecting steganography by looking at variances between bit patterns and unusually large file sizes”。隐写分析是发现、追踪和破坏隐蔽信息的艺术。与加密和解密一样,隐写分析的目标就在于识别可疑的信息流,检测在这些信息流中是否含有隐藏的信息编码,如果有就提取这些隐藏的秘密信息。隐写分析的挑战性在于:可疑的信息流中,如信号或文件可能并没有隐藏秘密信息。如果隐藏了信息,这些隐藏的秘密信息可能已经经过了加密处理才嵌入到载体信号或文件中。一些可疑的信息流可能只是噪声或不相关的数据嵌入了这些信息中。这种分析可能花费很长时间,除非有可能完全提取解密并且检查隐藏的数据,但可能只是怀疑信息流,不能确定这些信息流是否就是用于传输秘密信息。

隐写分析不同于加密分析。加密的数据很明显地就告诉截获者截获的信息流中包含了加密信息,但隐写分析只是认为这些数据流值得怀疑,但还不能确定它们是否包含秘密信息。所以要逐步得出可疑信息流的子集,分而化之,然后使用先进的统计技术来进行统计分析。那么要进行隐写分析需要知道信息到底可以隐藏到哪里呢?在开放的环境中,尤其是在 Internet 上,可以说可以隐藏的地方无处不在。

例如,在 Web 网页上有一系列的地方能隐藏信息。这些地方常出现在文本、图像、音频、视频等

- 文本 通过使用文本颜色与背景相同来隐藏文本,通过视觉的检测很难发现词和行间的很小的移动。为了发现可见的文本,按下 Ctrl+A 组合键对整个网页进行全选,为了检测文本是否被破坏,把整个网页复制粘贴到字处理器中。当然,秘密信息也可以隐藏到网页的上下文当中,检测这种嵌入秘密信息的方法就是分析最难的语法结构。
- 非文本元素 任何图片或媒体链接都可能隐藏链接或信息。
- 链接 链接的建立可以不使用下划线或者当鼠标指针经过这些链接时,链接不改变颜色。最简单的发现链接的方式是搜索 HREF=。另一种方式是使用 Tab 键来逐一移动光标到网页的所有链接。
- 注释 注释只有在网页的源代码中才能看到。
- 结构 许多浏览器都忽略没有经过说明的源代码信息,例如,通常在可选的标签中就可能有隐藏的线索。
- 框架 浏览在网页上每个框架的源代码。有时网站不能使用右击或者使用菜单功能时看不到源代码,在这些情况下,试着在地址栏使用如下命令:

```
view-source: http://(site url)
```

另外,最常见的就是将秘密信息隐藏到图片或者音频中。

无论是哪种方式,在电子媒体中使用信息隐藏可能引起媒体属性的改变,它可能引起某种性能的降低或者不常见特性的出现。在隐写图像中不常见的模式是最值得怀疑的。例如,有一些磁盘分析工具能过滤在存储设备中没有使用部分隐藏的信息。过滤器也能用于鉴别 TCP/IP 数据包,如果在包头中包含隐藏的或无效的信息。TCP/IP 数据包经常通过互联网进行传输,在数据包头中有未使用和保留的空间。数据包头是用户很少使用并进行读取的地方,所以这里是理想的隐藏数据的地方。使用这种方法的劣势在于防火墙可以配制成过滤到一些数据包,这些数据包的保留包头中包含不相称的数据。另外,在数据包头隐藏信息也是不可靠的。因为 TCP/IP 的头和保留的字段位有可能在路由处理时被覆盖,因此,导致隐写传输失败。通过隐写工具分析重复的模式也可以提示隐藏的信息。常见的就是将源图像与伪装图像进行分析比较,则可以看到显而易见的不同。这种方法称为 known-carrier attack。通过大量图像的比较,隐写工具的签名就会出现。

如果没有源图像可进行比较,从签名的起源也已能充分显示隐藏信息的存在以及鉴别嵌入信息所使用的隐写工具。对这类签名进行检测可以通过隐写分析工具自动检测。隐写检测利用调色板和签名来分析在某个区域中非常显眼的像素。另外一种可视的分析就是对图像进行填充和剪切。对于一些用隐写工具处理的伪装图像,没有填充的部分可能还是黑色空格,这样,在源图像和伪装图像在大小上会有所不同。另一种现象是某种特殊的颜色大量地增加或减少,或者在调色板中的颜色显著增加而不是随机增加(灰度图像除外)。但是优秀的隐写工具总能达到似是而非的特性。可以发觉有隐藏的信息,但是难于破解。如果发现的是音频或视频的可疑文件,而且有源文件,那么如果噪声增多,或者通过比较工具如 UNIX 的 diff 或者 Microsoft 的 fc,则可以很显著地比较出不同。但是如果沒有源文件,难度就会增大。对网络流量用统计分析可以提供最好的检测方法。如果统计分析表明数字图像与正常不同,什么是正常呢?在数字的音频、视频或图像中都包含一定的噪声。隐藏信息

正好是替代了这些噪声,每种类型文件都有源文件形式,所以对于每类文件都能产生一定程度的可预知的位分布方式,称为印迹(footprint)。对于不同的文件,都存在着不同,但是对于统计文件的内容,总存在着一定的可预知性,要么是比特位的随机性、平均信息量或者预定的模式。

仔细地选择适当的伪装图像和伪装工具,才能使隐藏信息不为人们所察觉。但是任何工具都会在图像上留下指纹或签名,这就表明这里有隐藏的信息。分离隐藏信息的第一步是信息分析。也可以认为是对隐藏信息的攻击。隐写分析包括检测、提取和破坏隐藏的信息。攻击方法依赖于使用了哪种隐写技术,也就是说,必须检测出是基于哪种隐写的信息流。针对隐写术的攻击有如下类型。

- 唯密写攻击(stego-only attack) 只有伪装对象可用于分析。例如,只有伪装载体和隐藏的信息可用。
- 已知原载体攻击(know cover attack) 原载体对象可以同伪装对象进行比较,可以进行模式识别。例如,原始图像和嵌入秘密信息的图像可以进行比较。
- 已知隐秘信息攻击(know message attack) 是已知隐藏信息的模式的分析,这将有助于隐写分析,即使有原始信息,对这些藏密对象的攻击仍是很困难、棘手的,其难度甚至不亚于 stego-only attack。
- 选择密写攻击(chosen stego attack) 隐写工具(或算法)以及伪装对象都已经知道,即攻击者拥有了工具(或算法)和伪装对象。例如,攻击者都已经知道软件和伪装载体。
- 选择密写结果攻击(chosen message attack) 藏密分析者将自己选择的信息使用藏密工具或算法产生伪装目标。这类的攻击目标在于检测伪装对象的相应模式,以便找到使用的特定工具或算法。
- 已知密写攻击(know stego attack) 隐写算法或工具都已经知道,并且源对象和伪装后的对象均可用。

因为隐写术与数字水印之间存在着差异,所以下面介绍一下数字水印的攻击类型。

- 擦除攻击(removal attack) 也称为简单攻击(simple attack)、鲁棒性攻击(robustness attack)、噪声攻击(noise attack)等。其目的是从含有数字水印的作品中移去或削弱水印,如某种信号(图像的亮度、对比度)的增强等。这类攻击包括滤波(如 wiener 滤波)、有损压缩或量化、统计平均、加噪、D/A 和 A/D 转换、重采样、重量化、模糊/反模糊、最大似然估计、邻块插值等。虽然并不是所有这些攻击手段都能够完全去除水印,但是它们却能对嵌入的水印信息造成极大的破坏。
- 表达攻击(presentation attack) 主要是破坏水印的检测环节。对含有水印的数据进行处理,使得水印检测器检测不到有效的水印,这就是表达攻击。采用的技术主要有:去同步攻击(desynchronization attacks)、Oracle 攻击等。
- 解释攻击(interpretation attack) 也称协议攻击(protocol attack),意图攻击水印应用的整个概念,使得提取的水印信息出现解释混乱,其中以 IBM 攻击和拷贝攻击最为典型。
- 共谋攻击 为了有效地保护版权,作品的发行商可能会为每个发行的拷贝嵌入不同的水印,以跟踪盗版者。在这种水印方案中,原始载体相同而水印不同。共谋攻击(collusion attack)就是多个用户利用同一原始载体数据的各自不同水印信号版本,

比较拷贝中的不同之处,从而可以找出拷贝中的部分指纹标记的位置。对这些位置上的标记进行修改(比如取简单平均值或加权平均值),从而生成一份新的近似载体数据,以此来逼近和恢复原始数据,其目的是使检测系统无法在这一近似的数据集合中检测出水印信号的存在。

- 法律攻击 数字水印还处在发展阶段,自身的理论、算法还需要进一步完善,相应公认的标准尚未建立,国际、国内的法律中对这种新生的信息安全技术尚未正式确认。因此,在应用数字水印进行版权保护时,攻击者可能利用法律条文中的空白来对水印进行攻击。这也是目前水印的大规模应用所面临的挑战。

第8章将进一步详细分析隐写分析技术。

## 1.8 信息隐藏当前研究现状与存在问题

目前,随着因特网的普及,以及信息处理技术和通信手段的飞速发展,使图像、音频、视频等多媒体信息可以在各种通信网络中迅速、快捷地传输,给信息的压缩、存储、复制处理等应用提供了更大的便利。同时,也为信息资源共享提供了条件,目前网络已经成为主要的通信手段。各种机密信息,包括国家安全信息、军事信息、私密信息(如信用卡账号)等都需要通过网络进行传输,但互联网是一个开放的环境,在其上传输的秘密关系着国家安全、经济发展和个人隐私等方方面面的安全,所以信息安全在当今变得越来越重要。

信息安全主要有两个分支:加密技术和信息隐藏。加密技术(cryptography)已经为人们所熟悉,广泛应用于各行各业。人们已经研究加密技术许多年,有许多加密方法,但是由于加密明确地告知用户,此文件或其他媒介已经进行过加密,窃密者必将利用各种破解工具进行破解,得到密文。虽然加密长度和强度一再增加,但所谓“道高一尺,魔高一丈”,破解工具也在加强。并且由于计算机性能的飞速发展,使解密时间缩短,所以加密术的使用局限性已见一斑。这使人们再一次转向了信息安全的另一个主要分支:信息隐藏。信息隐藏可以追溯到公元1499年,它的历史很久远。但是直到20世纪90年代在IT界,人们才赋予了它新的活力,使之成为继加密技术之后,保护信息的又一强有力的工具。信息隐藏与传统的信息加密的明显区别之处在于,传统的加密技术以隐藏信息的内容为目的,使加密后的文件变得难以理解,而信息隐藏是以隐藏秘密信息的存在为目标。所以科学技术的发展使信息隐藏技术在信息时代又成为新的研究热点。它既发扬了传统隐藏技术的优势,又具有了现代的独有特性。对于研究信息安全方向的学者而言,研究信息隐藏是很有意义的,也是刻不容缓的。

近几年国内许多学者也相继开展了信息隐藏方面的研究,国家有关科技发展部门也日益重视此方面的研究。国内信息隐藏的研究始于1997年,2003年后掀起热潮,CNKI收录的文献数量快速增长,并且主要集中于数字水印的研究。国内的一些大学(如北京大学、清华大学、哈尔滨工业大学、浙江大学、北京邮电大学和中山大学)和研究机构(如中国科学院自动化研究所的模式识别国家重点实验室和北京电子技术应用研究所)在该领域的研究与国外相比并不落后。1999年国家自然科学基金委员会政策局等在北京组织召开的“网络计算和信息安全论坛”,强调了研究信息伪装的重要性,建议在“十五”期间重点关注包括数字水印在内的网络环境下的信息安全领域的研究。国家973、863、国家自然科学基金重点课题中都对信息隐藏研究进行了重点资助。“十一五”期间,国家密码发展基金重点资助信息

隐藏、数字水印、可信计算等“新技术在密码研究中的应用”。

目前国内最具代表性的信息隐藏学术交流活动是全国信息隐藏研讨会(CIHW)。CIHW 是我国信息安全领域中信息隐藏及数字水印技术的专业学术交流活动, 我国信息安全领域的何德全、周仲义、蔡吉人等院士与有关应用研究单位联合发起了我国的信息隐藏学术研讨会并于 1999 年 12 月在北京召开了第一次会议。至今 CIHW 研讨活动已举行了七届全国会议。第六届全国信息隐藏学术研讨会(CIHW2006)于 2006 年 8 月上旬在哈尔滨工业大学召开, 第七届信息隐藏学术研讨会在南京理工大学召开, 第九届将于 2009 年在湖南大学召开。CIHW 对推动我国信息隐藏技术的研究与应用起到了积极的促进作用。

另外, 国内的“中国可信计算与信息安全学术会议”、“网络计算和信息安全论坛”也逐渐将信息隐藏作为会议的重要内容。我军目前有国防科技大学、信息工程学院、军械工程学院在 2002 年后开展了这方面的研究。

目前已有数家公司推出了各自数字水印的产品。成立于 2002 年的阿须数码利用数字水印实现了电子签章、数字水印条码和电子签章技术, 利用数字水印和数字签名技术, 结合 PKI 数字证书, 在电子文档上加上电子签章, 以保护电子公文及其衍生的纸质公文; 其数字水印条码技术解决了人们标识微小物品及表述附加商品信息的问题。北京握奇信、北京兆信和北京宇飞数字水印服务中心主要解决了数字水印技术在印刷扫描中的防伪应用。北京密安公司利用数字水印技术提出了节目信源版权保护、认证的方案。北京华旗数码影像研究院开发的爱国者 V80PLUS 数码相机加入了数字水印的嵌入式软件, 以解决数字照片的保真性, 但软件固化为芯片技术还没有完全解决, 2005 年 7 月进入国家 863 计划, 并被北京市版权局应用到北京数字作品版权登记平台。

现在信息隐藏研究方面还存在着几方面问题: 信息隐藏的基础理论研究还很薄弱, 不像加密技术一样有着广泛的理论基础; 并且有关的理论都尚未建立和完善; 隐蔽通信的容量计算和评估体系还尚未完善, 还有待于进一步研究和改善; 针对于隐蔽通信中的信道模型等方面还缺乏统一的模型, 商用的隐写系统的开发中还缺乏建模, 另外, 对于隐写容量的研究还待继续。作为信息隐藏的分支, 隐写术的研究目前也存在着上述问题, 都需要在进一步研究和开发中得到解决。

## 1.9 本章小结

本章是介绍性的章节, 在“引言”之后分别介绍隐写术、数字水印、隐蔽通信这三大信息隐藏分支, 然后对信息隐藏的应用和典型算法进行了概要性介绍, 并阐明了信息隐藏目前的研究现状, 同时给出了本书的整体框架。

## 1.10 因特网资源

在因特网和 Web 上有许多可用的资源, 例如:

- EzStego, Stego Online, Stego<<http://www.stego.com>>
- Steganos, Deus Ex Machina Communications<<http://www.steganography.com>>

- UnZign. Watermarking Testing tool <<http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/>>

## 1.11 复习题

- 1.1 什么是信息隐藏?
- 1.2 什么是隐写术?
- 1.3 什么是数字水印?
- 1.4 什么是隐蔽通信?
- 1.5 为什么在过去的几年中,信息隐藏变得如此受欢迎?
- 1.6 目前都存在哪些隐写算法?谈谈自己的理解。
- 1.7 现实生活中,你遇到过使用信息隐藏的哪些实际应用?
- 1.8 网络学习很重要,你知道哪些信息隐藏方面的因特网资源?

## 信息隐藏基本原理

### 本章目标

- 理解信息隐藏的基本原理。
- 了解隐写系统的分类。
- 理解一般的数字水印系统。
- 理解隐写术的主要术语和基本分类。
- 理解数字水印技术的主要术语和基本分类。

信息隐藏技术是近几年来国际学术界兴起的一个前沿研究领域。特别是在网络技术迅速发展的今天,信息隐藏技术的研究更具有现实意义。目前,为保证数据传输的安全,需要采用数据传输加密技术、信息隐藏技术、数据完整性鉴别技术。为保证信息存储安全,必须保证数据库安全和终端安全。信息安全的研究包括两个主要研究方向:信息加密与信息隐藏。在信息安全的研究理论体系和应用体系中,密码技术已经历了长期的发展,形成了较完整的密码学理论体系,有一系列公认的、经典的、可靠的算法。然而,在现代信息科学技术的条件下的信息隐藏,虽然可以追溯到公元前,但其完备的理论体系还尚未建立。本章将详细阐述信息隐藏的基本术语、基本原理等基础知识。

### 2.1 信息隐藏的基本原理与分类

Lisa M. Marvel 博士于 1999 年提出了扩展频谱的图像信息隐藏(spread spectrum image steganography, SSIS),阐述了信息隐藏的基本原理,那就是先将秘密信息嵌入到噪声信号中,然后再随噪声信号嵌入到数字图像中,因为数字图像中存在着噪声信号,如果噪声的容量不大,在没有原始图像比较的情况下,一般人类视觉和计算机统计分析都无法感知到秘密消息的存在,这样隐写就成功了。在提取秘密信息时,需要图像恢复和差错控制编码技术,前者得到原始图像的近似估计,并对嵌入的秘密信息容量进行估计,然后根据相应的算法进行提取。这种方案的位错率较高,所以必须对秘密信息也进行低比率差错编码,才能较完善地提取秘密信息。

目前,隐写术的基本原理可以概括为:首先,对欲嵌入的秘密信息进行预处理,预处理包括加密、编码然后生成伪随机数,再将预处理后的秘密信息根据相应的嵌入算法嵌入到载

体中,载体可以包括文本、图像、语音和视频等多种格式的多媒体,在通信中可以使用隐蔽信道进行通信,最后在提取中根据相应的提取算法和密钥提取秘密消息,这样,就可以达到三层安全。对相应的嵌入算法和提取算法都要分析不可感知性、容量和鲁棒性三者之间的关系,理论上使三者之间平衡并性能达到最佳。

信息隐藏的主要分支如图 2.1 所示。

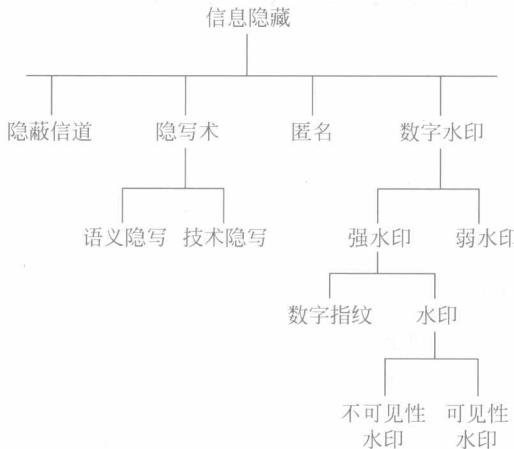


图 2.1 信息隐藏的主要分支

根据分类的依据不同,可以将隐写术分为以下几类。

- 按隐写系统结构分类 分为纯隐写术、密钥隐写术和公钥隐写术;
- 按隐写空间分类 可以分为空域隐写、变换域隐写;
- 按隐写载体分类 可以分为文本隐写、图像隐写、音频隐写、视频隐写和二进制隐写。

下面我们分别介绍相应分类中隐写术的实现。

### 2.1.1 纯隐写术、密钥隐写术和公钥隐写术

先分析理论上完美的隐蔽通信,隐写术将包含在其中。为了阐明这个概念,现举例说明:假设由三个人分别为 Alice、Bob 和 Denmy。Alice 想发送秘密信息(M)给 Bob,使用无害消息作为载体(C),然后秘密消息嵌入到 C 中形成伪装载体,伪装载体将发送给 Bob 并且没有引起任何怀疑,这是纯隐写的过程;或者,Alice 使用隐写密钥(stego-key(K))将秘密消息(M)嵌入到载体(C)中生成伪装载体(S)。然后 Alice 将伪装载体(S)发送给 Bob 而没有引起 Denmy 的任何怀疑。而 Bob 能够阅读秘密信息,因为 Bob 有相同的隐写密钥(stego-key(K));但在通信双方必须对密钥进行协商,达成一致后,双方才能进行通信,这是密钥隐写过程;如果在这个过程中 Alice 在嵌入秘密信息时使用的是公钥,而 Bob 在提取信息时使用的是与公钥为一对密钥的私钥,那么这是公钥隐写过程,在公钥体制中,为了防止篡改,必须要借助于证书体系来验证公钥。

无论是哪种秘密通信,从理论上隐蔽通信是很完美的,就如 Fabien A. P. Petitcolas 所指出的:“在完美的系统中,无论是人或者计算机通过统计分析都不能区分出是一般载体还

是伪装载体。”但在实际中,情况并非总是这么完美。为了将秘密消息嵌入到载体中,载体必须包含足够的冗余数据或噪声。这是因为隐写术的嵌入处理过程中实际上是使用秘密消息替换这些冗余信息。隐写术的框架如图 2.2 所示。

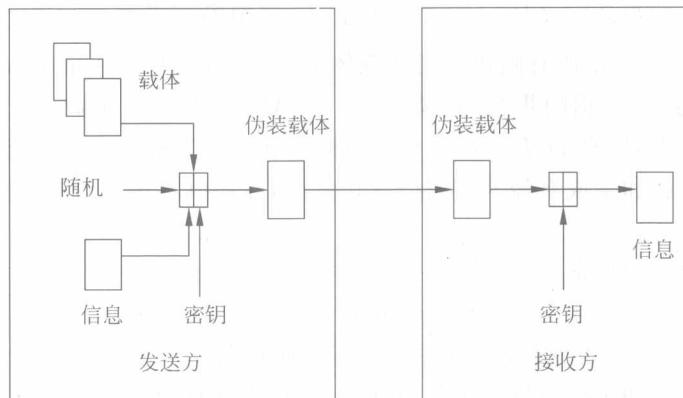


图 2.2 隐写术框架

数学上发送方嵌入的过程和接收方提取的过程都可以通过映射描述:前者为  $E: C \times M \rightarrow C$ ;后者为  $D: C \rightarrow M$ ,而且  $|C| \geq |M|$ (其中,  $E$  为嵌入过程,  $D$  为提取过程,  $C$  为载体,  $M$  为秘密信息)。

下面分别介绍纯隐写、密钥隐写和公钥隐写术。

纯隐写的文字定义为:“一个不需要交换隐写密钥的隐写系统。”这种类型的隐写安全性最低,因为在这种隐写中,通信双方只能假定没有任何第三方能察觉发送的秘密消息,但是使用开放的环境,如在互联网上,这种情况绝对不会发生,网络上的任何通信都可能被第三方截获。如果使用纯隐写,秘密消息被隐写分析出来的可能性最高。因为只要第三方知道嵌入算法,就可以提取出相应的秘密信息。

纯隐写的数学定义如下:

**定义 1(纯隐写系统):** 对一个四元组  $\sum = \langle C, M, D, E \rangle$ ,  $C$  是所有随机选择的载体的集合,  $M$  是所有可能嵌入的秘密信息的集合,且满足  $|C| \geq |M|$ ,  $f$  是嵌入函数,  $f^{-1}$  是提取函数;对  $m \in M$  和  $c \in C$ ,恒有  $D(E(cm)) = m$ ,则称该四元组为纯隐写系统。

纯隐写系统的原理图如图 2.3 所示。

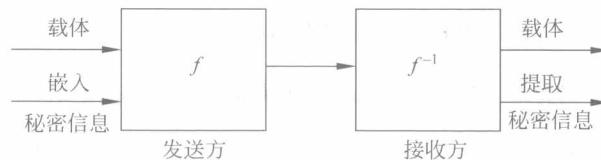


图 2.3 纯隐写系统原理图

密钥隐写系统文字定义为:“a steganographic system that requires the exchange of a secret key(stego-key) prior to communication”。也就是说在双方进行秘密通信前,需要双方隐写系统交换密钥。密钥隐写使用密钥将秘密消息嵌入到载体信息中,只有知道密钥的

人才能执行相反的过程,在载体中提取秘密消息。与纯隐写不同,纯隐写通信中就像存在一条无法感知的通信信道。而密钥隐写需要交换密钥,这就可能引起第三方的截获或怀疑。而密钥隐写的优势在于即使隐写消息被截获,只有知道密钥的用户才能提取秘密信息。

密钥隐写系统的数学定义是一个五元组定义(密钥隐写系统):对一个五元组  $\Sigma = \langle C, M, K, D_K, E_K \rangle$ ,  $C$  是所有随机选择的载体的集合,  $M$  是所有可能嵌入的秘密信息的集合,  $K$  是所有可选择的密钥的集合,且满足  $|C| \geq |M|$ ,  $f$  是嵌入函数,  $f^{-1}$  是提取函数;对  $m \in M, c \in C$  和  $k \in K$ , 恒有  $D(E(cm)) = m$ , 则称该五元组为密钥隐写系统。

因为密钥的算法和生成有经典的密码学作基础,在此就不再多讨论,这已经超出了本文的范围。

公钥隐写采用公钥加密。公钥隐写定义为“a steganographic system that uses a public key and a private key to secure the communication between the parties wanting to communicate secretly”。也就是说,隐写系统需要使用公钥和私钥来确保双方通信的隐蔽性。发送方在嵌入处理过程中将使用公钥进行,而只有在拥有了私钥的接收方才能解密并提取秘密信息。公钥和私钥是同时生成的,在数学上有直接的联系。公钥体系有广泛的使用基础,也是一种很成熟的技术,在目前都借助于国际认证的 CA 颁发的数字证书来验证,公钥可以在网上公开发行,私钥由申请证书的用户收藏。用公钥加密的信息,只有私钥能解开,而用私钥加密的信息,也只有公钥能解开。在隐写术中使用公钥这项技术,无疑推动了隐写术更加广泛的使用。在公钥为基础的隐写术中,能达到几层安全,首先第三方必须怀疑已经使用了隐写术,而且必须使用公钥体系来发现相对应的破解算法,然后才能提取秘密消息。这种处理方式的隐写在三种方式中是最为安全的。因此,隐写术并没有脱离加密技术,它们可以说是孪生兄弟,所以这两种技术的结合,可以使双方的优势互补。不同之处在于密钥(Key)不同。

密钥隐写和公钥隐写的原理图如图 2.4 所示。

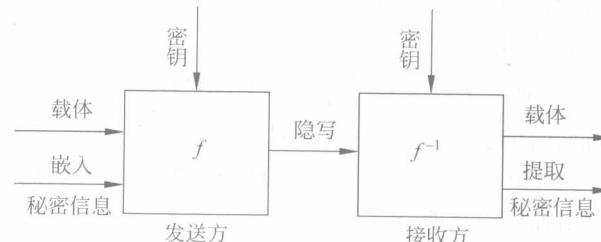


图 2.4 密钥和公钥隐写原理图

关于空间域和变换域的隐写,将在第 4 章中详细介绍,下面介绍另外一种根据载体不同而进行分类的隐写。

### 2.1.2 文本、音频、图像的隐写

目前隐写术使用的载体分别为文本、音频、图像和视频以及相应的二进制多媒体。下面分别介绍在各种载体中隐写术实现的基本原理。

首先介绍如何在文本中隐藏秘密信息。

给秘密信息编码并嵌入到文本文件中,这是一项具有挑战性的工作,因为文本文件中可供秘密信息替代的冗余数量有限。另外一个显著的缺点就是在第三方改变文本,或者将文本转换成其他格式(如从.txt到.pdf)时,基于文本的隐写就会被发现,导致隐写失败。但是还是有许多方法来完成文本中的隐写。下面介绍比较流行的编码方式。

- 行移编码法(line-shift encoding) 调整文本文件中的垂直行距来隐藏信息。具体的是选择特定的段落,将其行距作垂直的上或下调整,根据人类视觉的特点,行距值必须在3cm以下才能不被察觉。
- 字移编码法(word-shift encoding) 它的使用方式与行距调整方式相同,在此使用文件中的水平方向位置来嵌入秘密信息。具体的是选择单个字进行左右调整,然后嵌入秘密信息,字距调整法的隐蔽性比行距调整法强,但它需要文本的格式支持变化的字间距。
- 特征编码法(feature specific encoding) 它是通过改变特定文本的属性来嵌入秘密信息的编码,如每个字符水平/垂直的长度。这是最难被第三方察觉的编码,因为对每种类型格式的文件都有许多特征可以改变用于为秘密消息编码,但使用时要注意字体等特性的修改文件内容是否改变等。

上述介绍的三种方法都需要源文件或者源文件的格式才能在接收方进行解码。因为基于文本的隐写很容易被觉察,所以文本中的隐藏目前使用受到限制。

下面介绍比较常见的图像为载体的隐写。

将秘密信息编码嵌入到数字图像中是目前使用最广泛的一种隐写。因为它利用人类视觉(human visual system, HVS)有限性的特性,就是说对数字图像的某些区域,人类视觉不敏感,并且图像中发生微小变化,人眼看不到改变等。绝大多数的文本、图像或密文及其他任何形式的媒体都可以生成比特流嵌入到数字图像中。随着数字图像广泛的使用,载体为数字图像的隐写也在持续增长。以图像为载体的隐写系统如图2.5所示。

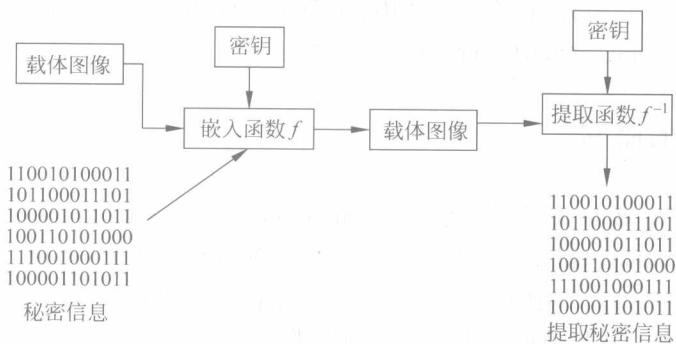


图2.5 图像为载体的隐写系统

对于计算机,图像就是用许多代表亮度的像素的阵列组成,这些像素组成了图像的光栅数据。当使用图像作为隐写术的载体时,通常使用8位或者24位的图像。每种载体都有其优势和劣势。8位图像因为其相对较小的尺寸正在广泛使用,但是缺点是在编码中它可以使用的位数只有256位。通常,当处理8位图像(如.gif)时都使用灰度图像调色板,因为这

类图像颜色上的逐渐变化在源图像和嵌入秘密信息的伪装图像之间很难进行区分。24位图像更适用于隐写。它可用于编码的位数超过了1600万。这个数字远远超出了人的视觉范围。这就使人很难察觉到秘密消息的存在。另外一个优势在于嵌入的秘密消息的容量可以增大,也就是说可以比8位图像嵌入更多的秘密消息。当然,所有嵌入信息容量都要比载体小。但8位通常嵌入的信息量使用的单位是KB,而24位图像使用的是MB。如果图像太大,不利于在Internet上传输,所以在数字图像中广泛使用着两种压缩技术:有损压缩和无损压缩。有损压缩(如JPEG)经常用于24位图像来减小它的大小,但是对于伪装图像,经过有损压缩之后,嵌入的秘密信息也可能丢失。因为秘密信息是替代图像的冗余部分,所以可以去掉。而无损压缩,对源图像没有任何损失。这也是隐写选择无损压缩的原因。如对.bmp和.gif格式的图像,但这种压缩并不能很好地减小图像的大小。

隐藏信息的大小必须比图像小,而像素多少代表了文件大小。通常图像都为 $640 \times 480$ 像素,数据大小为307200比特,通常高质量的24位的图像大小是 $1024 \times 768$ ,数据大小为2359296比特,24位的图像可以在每个像素中嵌入3位数据,因为它使用每个像素的三位来表示颜色值。GIF文件通常使用8位调色板,因此只能表示256种颜色。JPEG文件通常使用24位调色板,而BMP图像可以使用8位也可以使用24位调色板。24位图像为隐藏信息提供了更大的可选空间。图像一般都使用中间色或一些过渡,由于图像的这种特性,许多信息都可以隐藏到图像中,并且嵌入信息的图像变化人眼是不能察觉的。在图像嵌入编码中,经常使用最低有效位法和伪装与过滤技术。LSB使用图像中的每个像素来隐藏信息,对于24位图像,可以选择每个像素中的3位作为隐藏位,对于8位图像,每像素中使用1位来隐藏数据。由此可见,24位图像可以隐藏更多的信息。但是根据载体图像的彩色调色板,如果每像素嵌入2位最低有效位,人类视觉还不能分辨源图像和伪装图像的差异,这种技术存在的问题是当图像或图像格式发生改变时,隐藏失败。伪装和过滤技术更多用于有损压缩技术,这种技术实质是扩展了图像,将秘密信息伪装成图像的数据,使之成为源图像的扩展部分,一些专家指出,这是一种很好的信息隐藏技术,但不属于技术隐写,这种技术就是鲁棒性非常出色。另外,在隐写过程中,还会使用许多复杂的算法、图像传输技术和图像加密技术,将在后面的章节中进行介绍。

### 2.1.3 音频中的隐写

音频中的隐写是根据人类听觉系统(human auditory system, HAS)来进行的,因为众所周知,对相同频率的音频信号,人与人之间的敏感度有很大差异,所以在隐写术中对音频进行编码是具有挑战性的一项工作。听觉系统中存在一个听觉阈值电平,低于这个电平的声音信号就听不到,听觉域值的大小随声音频率的改变而改变,各个人的听觉域值也不同。大多数人的听觉系统对 $2\sim5\text{kHz}$ 之间的声音最敏感。一个人是否能听到声音取决于声音的频率,以及声音的强度是否大于该频率对应的听觉阈值。因为人类听觉系统是一个动态的范围。Boney等人根据人类听觉系统的这一特性,将秘密信息隐藏到强度较弱的频率中,也就是说,某段声音频率的强度之上,人能听到,这一强度之下,人就不能听到这段声音,那么,就可以将相应的时间轴上的信号转换到频率轴上,计算出各频率的强度,然后将秘密信息嵌入到比这些频率强度低的各频率中去。根据听觉掩蔽特性,也就是说声音的响度不仅

取决于自身的强度和频率,而且也与同时出现的其他声音有关。各种声音可以互相掩蔽,一种声音的出现可能使得另一种声音难以听清。一种频率的声音阻碍听觉系统感受另一种频率的声音的现象称为掩蔽效应。前者称为掩蔽声音(masking tone),后者称为被掩蔽声音(masked tone)。Flanagan 等人则是根据人对音强敏感度不同,将秘密信息加载在到较高强度的不同比特中。

频域掩蔽,也就是说一个强纯音会掩蔽在其附近同时发声的弱纯音,这种特性称为频域掩蔽。例如,同时有两种频率的纯音存在,一种是 1000Hz 的声音(60dB),另一种是 1100Hz 的声音(42dB),在这种情况下,1100Hz 的声音就听不到。弱纯音离强纯音越近就越容易被掩蔽。不同纯音的掩蔽效应曲线,例如,在 250Hz、1kHz、4kHz 和 8kHz 纯音附近,对其他纯音的掩蔽效果最明显,低频纯音可以有效地掩蔽高频纯音,但高频纯音对低频纯音的掩蔽作用则不明显。Gruhl 则根据这一特性,利用不同长度的回音,先计算声音的延迟,然后生成回音,然后对回音进行处理,使其强度低于人能听到的范围,然后将源音乐划分成数个时间区,把回音加到各个区段中。在使用的音频格式中有三种主要的数字格式:取样量化(sample quantization),临时采样率(temporal sampling rate)和知觉采样(perceptual sampling)。像 WAV 和 AIFF 格式这样广泛使用的音频使用的就是 16 位线性采样结构的取样量化。临时采样率用于可选的通常在 kHz 的频率段进行采样。通常采样率越高,可以使用的用于隐藏信息的空间越多。

知觉采样很大地改变了音频的格式,它只对可听到的音频部分进行编码,这种编码虽然音频还在,但改变了信号,这种格式在 Internet 上广泛使用,如 MPEG(MP3)。对于音频的传输介质,W. Bender 指出需要遵循以下四点:从一端到另一端的传输过程中,数字信号没有改变;提高/降低采样频率是可以的,但必须保留所有数字信号;当重采样时,不同采样频率的信号也随之变化;通过空气传输时,应以无线电频率传送。在音频的隐写中使用三种很流行的编码。最低位编码(low-bit encoding)、相位编码(phase-encoding)和扩频(spread spectrum)。最低位编码是将秘密消息嵌入到音频文件的最低有效位中。信道容量是 1KB 每秒每 kHz,也就是说,44kB/s 对应着 44kHz 的采样序列。这种方法虽然很容易实现,但是当信道噪声增加或重新采样时,很容易造成数据损失。相位编码用表示隐藏数据的相位来代替源音频文件的相位。常用的有离散傅里叶变换(discrete fourier transform,DFT)。扩频几乎对所有频谱都进行编码,然后将音频通过不同的频率进行传输,直接序列扩频(direct sequence spread spectrum,DSSS)是这样一种技术,用高速伪随机码将传输信息所需带宽加以展宽的一种扩频技术。若传送的信码速率为  $R_b$ ,用速率为  $R_c = MR_b$  伪随机码与传送信码作模二加后,再进行相移键控,频带就展宽  $M$  倍,将信号功率分散到较宽频带内,类似于噪声,不易被发现,对其他通信系统的影响也小。扩频用于音频中的隐写相对比较安全,但它相对于原音频而言,引进了噪声,这样可能引起数据的丢失。

在 Internet 网上,每个站点几乎都有音频媒介。秘密信息可以嵌入这类媒体。最流行使用的媒体的格式是.wav 和.mp3。下面是将密钥嵌入到.wav 和.mp3 音频文件的示例,分别如图 2.6(a)和(b)所示。

隐写术的编码方法多种多样,都存在着各自的优点和缺点,因载体不同,选择的方法也会不同。上述介绍了各种编码特点以及相应的基本原理,下面介绍隐写术的主要术语。

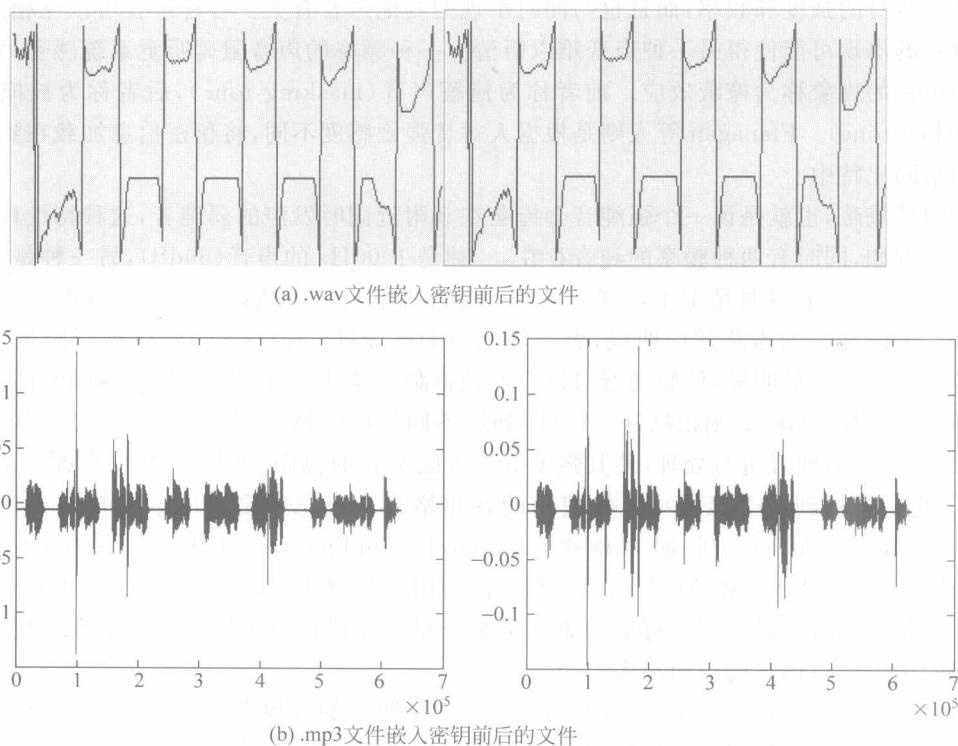


图 2.6 .wav 文件和.mp3 文件嵌入密钥前后

## 2.2 信息隐藏的主要术语

本节将根据信息隐藏的分类对信息隐藏的术语进行归纳和总结,以期达到一致。

- **信息隐藏(information hiding)** 作为信息安全的一个重要分支,主要是将秘密信息嵌入到其他载体中不让未授权的第三方察觉的一门学科。也就是说将秘密信息嵌入到多媒体数字信号本身所存在的冗余中,而且不影响载体信号的感觉效果和使用价值。它主要包括隐写术、数字水印、隐蔽通道、匿名通信等分支。一般而言,信息隐藏技术的研究及应用主要是指隐写术和数字水印两个领域。
- **匿名通信(anonymous communications)** 匿名通信定义了一个经过多个中间结点转发数据的多级目标路径,为隐蔽接收者,发送者可选定  $N$  个连续目标,其中之一为真正接收者。窃听者在一段链路上获取真正接收者的概率为  $1/N$ ,并且中间结点在传送消息时可采取重新排序、延迟和填充手段使获取真正目标的概率更低,从而加大攻击者进行流量分析的难度。所选目标能可靠地完成上述工作且彼此间存在安全通道;中间结点必须知道整个路径结构才能重新排序。信息隐藏中的匿名技术就是设法隐藏消息的来源,即隐藏消息的发送者和接收者。例如,收发信者通过利用一套邮件转发器或路由器,就能够实现掩盖消息痕迹的目的,只要这些中间环节相互不串谋。因此,剩下的是对这些手段基础的信赖。需要注意的是,不同的情

况决定谁要“被匿名”，是发信者，还是收信者，或是两者皆要。网上浏览等将问题集中于收信者的匿名，而电子邮件用户关心的是发信者的匿名。

- 隐蔽信道(covert channel) 是指这些通道一般被运用于不可信程序，当对别的程序执行操作时，将有关信息泄露给不可信程序的拥有者，而不是指这些信道平时是隐蔽的(不可见的)。隐蔽信道一般存在于多级保密系统的背景之中。在操作系统中隐蔽信道是指使两个共同执行的进程，以违反系统安全策略的方式传输通信信道。隐蔽信道分为隐蔽存储信道 (covert storage channel) 和隐蔽定时信道 (covert timing channel) 和国下信道，前两种是常指的隐蔽信道。隐蔽存储信道是采用特殊的编译码方式使不合法的信息流逃避常规的安全控制机构的检测来实现。如在操作系统中是指由一个进程直接或间接写一个存储地址，而由另一个进程直接或间接读一个存储地址的隐蔽信道。隐蔽定时信道是利用时间轴上的事件序列进行编码来实现，在操作系统中一个进程通过调整自身对系统资源(如 CPU 时间)的使用，向另一进程发送信息。国下信道是指基于公钥体制的数字签名、认证应用等输出密码数据建立起来的一种隐蔽信道。除接收者之外，其他第三方均不知道密码数据中是否有国下消息的存在。
- 隐写术(steganography) 就是将秘密信息嵌入到载体中，而使伪装载体在人类视觉以及计算机分析时秘密信息不被发现，并且源载体与伪装载体之间差异很小，一般感觉不到载体的变化。一般把隐写术分为技术隐写、语义隐写。现在的隐写术主要利用高空间频率的图像数据隐藏信息，采用最低有效位方法将信息隐藏到宿主信号中，使用信号色度隐藏信息方法，在数字图像的像素亮度的统计模型上隐藏信息的方法，patchwork 方法等。
- 数字水印(digital watermarking) 数字水印主要是向被保护的数字对象嵌入某些能证明版权归属或跟踪侵权行为的信息。目前主要有两类数字水印，一类是空间域数字水印，另一类是频率数字水印。空间数字水印的典型代表是最低有效位算法，其原理通过修改表示数字图像的颜色或颜色分类的位平面，调整数字图像中感知不重要的像素来表达水印的信息，以达到嵌入水印的目的。频率数字水印的典型代表是扩频算法，其原理是通过时频分析，根据扩展频谱特性，在数字图像的频率域上选择那些对视觉最敏感部分，使修改后的系统隐含数字水印信息。
- 秘密信息(embedded data) 欲嵌入的信息。秘密信息是指掩藏在公开信息中的保密信息，也即发信者真正想要发送给收信者而又不想让未授权第三方知道的信息。
- 载体(cover) 是指秘密信息嵌入的对象，也就是用于容纳秘密信息的载体。
- 伪装载体(stego-object) 将秘密信息嵌入到载体之后形成的目标载体，称为伪装载体。也就是说，此时秘密信息已经隐藏在载体之中。
- 伪装密钥(stego-key) 无论使用私钥还是公钥体制，都可以使用密钥对嵌入的秘密信息进行加密处理来多层保护欲隐藏的信息。
- 隐写分析者(steganalyst) 隐写分析者就是检测分析隐蔽信息存在的真实性并通过各种隐写分析手段得出隐藏的秘密信息。隐写分析者分为主动攻击者和被动攻击者。主动攻击者不仅要检测嵌入秘密信息的存在，并且要破坏秘密信息甚至在检测出的秘密信息中嵌入自己的信息。而被动攻击者只是检测分析秘密信息是否

存在。

- 被保护信息(cover data) 数字水印技术中的伪装载体虽然也作为掩护水印信息，但它主要目标是保护水印信息。
- 水印密钥(watermarking key) 控制水印嵌入隐藏过程的密钥。水印密钥空间需足够大，而且分布比较均匀，即使第三方知道了水印嵌入算法的全部细节，但不知道秘密密钥，就不能将水印提出或破坏。水印体制的商业应用，其算法必须公开。所以数字水印算法的安全性完全取决于密钥，而不是以算法的保密来取得安全性。为了给攻击者增加去除水印的难度，目前大多数水印制作方案都在加入、提取、检测时采用了一个或多个密钥，做到只有掌握密钥的人才能读出、提取水印。
- 含水印信息(watermarked data) 被水印标注了的数据。
- 攻击者(attacker) 数字水印攻击者的目的是检测出数字水印的存在事实(在不可感知水印中)并破坏水印信息，其侧重点是盗用、破坏、删除、修改水印信息。数字水印的攻击者一般为主动攻击者或恶意攻击者，其主要攻击行为是检测数字水印的存在事实(在不可感知水印中)并盗用水印信息(盗版)，破坏、删除、修改水印信息及加入自己的水印信息。

在上述的常见术语介绍完成之后，还需要介绍信息隐藏的问题空间。因为信息隐藏的问题空间最终可以归纳为容量和不可感知性之间的关系。为了防止载体信号的质量下降，信息隐藏算法不能嵌入大量的隐藏信息，而大量信息的引入必然增加对原载体内容的更改，造成不可感知性的下降。而鲁棒性和不可感知性也存在着这样的矛盾，所以对嵌入率或者对格式的更改都有严格限制。当容量增加，不可感知性必然有不同程度的下降，而不可感知性的增强，必然造成容量的下降。所以这几个方面都限制着信息隐藏的处理过程。既要考虑隐藏数据的数量，又要使嵌入的数据使载体图像不失真，并且隐藏的数据必须达到不可感知性，对第三方的剪切、修改或删除等具有鲁棒性。

## 2.3 数字水印系统的构成与分类

在知识产权保护中应用最广泛的就是数字水印系统，它能提供所有者的身份，能进行所有权验证，并能对操作进行跟踪，例如，DiVX 公司生产的 DVD 播放器的安全技术之一就是使用了操作跟踪的设计水印。数字水印还能对内容进行认证和拷贝控制。由于数字水印与隐写术一样均属于信息隐藏领域的分支，有共享的技术，但两者之间存在着差异，在本节中主要针对数字水印这一分支的构成原理及特性进行分析和介绍。

数字水印(digital watermark)技术是指用信号处理的方法在数字化的多媒体数据中嵌入隐蔽的标记，这种标记通常是不可见的，只有通过专用的检测器或阅读器才能提取。数字水印是信息隐藏技术的一个重要研究方向。

嵌入数字作品中的信息必须具有以下基本特性才能称为数字水印。

- 隐蔽性 在数字作品中嵌入数字水印不会引起明显的降质，并且不易被察觉。
- 隐藏位置的安全性 水印信息隐藏于数据而非文件头中，文件格式的变换不应导致水印数据的丢失。
- 鲁棒性 所谓鲁棒性是指在经历多种无意或有意的信号处理过程后，数字水印仍能

保持完整性或仍能被准确鉴别。可能的信号处理过程包括信道噪声、滤波、数/模与模/数转换、重采样、剪切、位移、尺度变化以及有损压缩编码等。

在数字水印技术中,水印的数据量和鲁棒性构成了一对基本矛盾。从主观上讲,理想的水印算法应该既能隐藏大量数据,又可以对抗各种信道噪声和信号变形。然而在实际中,这两个指标往往不能同时实现,不过这并不会影响数字水印技术的应用,因为实际应用一般只偏重其中的一个方面。如果是为了隐蔽通信,数据量显然是最重要的,由于通信方式极为隐蔽,遭遇敌方篡改攻击的可能性很小,因而对鲁棒性要求不高。但对保证数据安全来说,情况恰恰相反,各种保密的数据随时面临着被盗取和篡改的危险,所以鲁棒性是十分重要的,此时,隐藏数据量的要求居于次要地位。

### 2.3.1 数字水印系统

数字水印技术是指将特定的信息,像所有者、商标、数字签名嵌入到载体中,来证明对载体的所有权等。载体信息可以是任何多媒体数据。数字水印系统包括水印的嵌入和检测、提取过程,数字水印系统的一般构成如图 2.7 所示。图 2.7(a)为水印嵌入,功能是将水印嵌入到载体数据中;图 2.7(b)为水印检测,功能是判断是否含有指定的水印并对水印可信度进行测量。

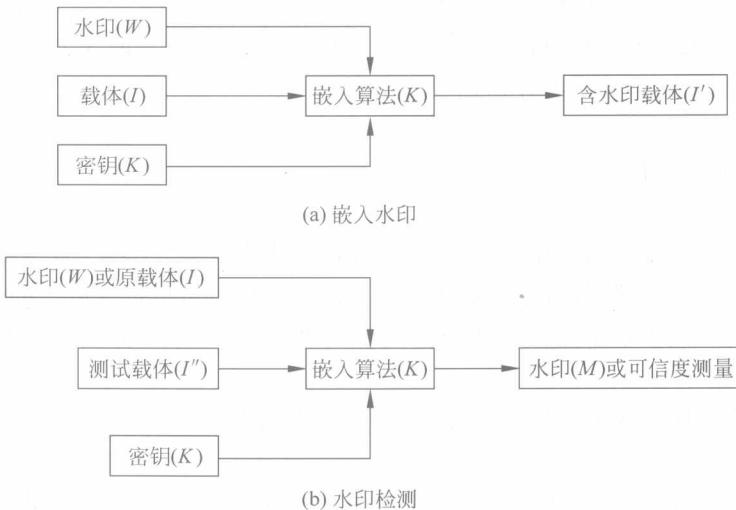


图 2.7 数字水印的一般系统

在数字水印的一般系统中,可以将数字水印统一表示为  $W = \{\omega(k) \in U, k \in W^d\}$ ,  $W^d$  表示水印域。典型水印模型可以用六元组表示  $\{I, W, K, E, \xi, D\}$ ,  $I$  表示载体(即被保护多媒体)的集合,  $W$  是水印的集合,  $K$  是水印密钥的集合,  $E$  表示用密钥  $K$  与载体  $I$  生成的水印的算法,  $E: I \times K \rightarrow W$ ,  $W = E(I, K)$ , 水印嵌入算法将在第 6 章详细讲解。 $\xi$  表示将水印嵌入的算法,  $\xi: I \times W \times K \rightarrow I$ ,  $I' = \xi(I, W)$ , 一般还可以描述为:  $I'(k) = I(k) \oplus h(k) \oplus w(k)$ , 也就是嵌入有加法规则和乘法规则。 $D$  为检测算法,  $D: I \times K \rightarrow \{0, 1\}$ , 可以用二值判定来判断水印的有无。

### 2.3.2 数字水印、隐写术与加密术的区别

加密是保护消息的内容,隐写术是隐藏信息的存在,我们在1.2节中已经详细讨论了隐写术。而水印非常强调针对各种攻击所具有的鲁棒性,而且并不总是将版权水印进行隐藏,有时一些数字水印系统使用的是可见数字水印,但通常指的水印都是不可见水印,不可见水印应用的范围非常广泛。常见的可见水印一般都是视觉模式,如公司标识或版权标记出现在数字图像之上,来指明数字图像的所有者。数字水印和隐写术作为信息隐藏的最重要的两类应用,可以使用相同的技术但也存在着差异,针对隐藏目的而言,数字水印是为了证明载体的版权所有或验证其完整性;隐写术则是为了将隐体信息通过公开的载体信道秘密传递给接收方。针对载体和隐体而言,数字水印的载体是特定的数字作品,隐体是特定的版权标识或作品摘要,二者对攻击者而言目标是确定的;隐写术的载体是类似随机的,越普通越具有隐匿性,而且隐体的类型是不确定的。针对保护对象而言,数字水印中隐体的存在是为了保护载体;隐写术中,载体的存在是为了掩护隐体信息的传输。针对隐藏方法而言,鲁棒性数字水印要求隐体的嵌入是鲁棒和难以去除的;信息隐匿则只要能保证其隐秘和安全,不关心鲁棒性。针对隐藏时效性而言,数字水印要求隐体的存在是长效的;而隐写术则只要求隐体在隐匿通信过程中存在即可,基本是一次一隐。针对隐藏容量而言,数字水印只要能证明版权或验证完整性即可,对容量要求不大;而隐写术可能需要传输的是电子文档或军事情报,容量要求较大。针对提取要求而言,数字水印允许一定的误差存在;隐写术则因为隐体信息的重要性,不允许有任何提取错误发生。

### 2.3.3 数字水印的分类

针对于各种不同类型的水印技术,数字水印的分类方式有许多种,因为数字水印系统涉及不同的领域(主要是信号处理和计算机安全)和背景,所以可以按照嵌入的载体、嵌入的方法、算法的鲁棒性、嵌入水印是否可见、嵌入水印个数的多少、嵌入算法是否可逆或无损等不同指标进行分类。

数字水印依据所嵌入的载体进行分类,主要分为图像水印、音频水印、视频水印和文本水印等。

按照水印嵌入域,数字水印可分为空域水印、变换域水印和压缩域水印。

按水印抗攻击能力分类,图像数字水印可分为鲁棒性水印、脆弱性数字水印、半脆弱性水印。

按嵌入的方法可以分为全盲水印、半盲水印和非盲水印。

按嵌入的水印是否可见,可分为可见水印和不可见水印。

按水印是否可逆,可分为可逆水印和不可逆水印。

按水印是单个还是多个,可分为多重水印和单水印。

按照所选水印的意义,可分为无含义水印和有含义水印。

另外还有公开水印、私密水印和对称水印,这种分类方式是根据在嵌入阶段,生成随机序列时所采用的Key类型不同而划分的,如是公钥则称为公开水印,如是私钥则称为私密

水印,如果采用的是对称密钥,则称为对称水印。

因为使用不同的技术和算法,所以水印的分类大致如上所述。下面介绍数字水印的特性。

### 2.3.4 数字水印的特性与术语

数字水印系统一般要具有一定的特性才能保证数字产品版权保护和完整性鉴定。数字水印系统一般的特性如下所述。

(1) 不可感知性。在数字水印中,很注重高保真度,也就是说嵌入水印的载体与原始载体必须非常接近,水印的嵌入不能引起宿主媒介质量的很大变化。

(2) 鲁棒性。并不是所有的数字水印系统都要求这个特性,只有鲁棒性水印系统要求,将对宿主进行空间滤波、有损压缩、打印、扫描以及旋转、平移等各种操作时,水印不会去除。

(3) 安全性。数字水印系统应该对非法提取具有很强的免疫力,对抗未授权的删除、嵌入和检测,从而保护数字产品。一般都使用密码术的经典算法来保证密钥的安全。

(4) 密钥的唯一性。不同的密钥应产生不同的水印。

(5) 嵌入的有效性。嵌入的水印能提取出来的概率。

(6) 虚警率。实际不含水印的产品检测出有水印的概率。

数字水印系统中常用的术语如下所述。

(1) 宿主信号(cover-signal)。也就是想嵌入秘密信息的载体信号,一般指音频、图像、视频等可视信号。

(2) 水印。加入到宿主信号中的秘密信息。

(3) 有效载荷(payload)。指在单位时间内水印编码的比特数。

(4) 水印访问单元(watermark access unit)。宿主信号中的最小部分,在这个单元可以可靠地检测并提取有效载荷。

(5) 容量(capacity)。水印访问单元能负载的有效载荷的比特数量。

(6) 水印方案(watermarking scheme)。嵌入和提取所需要的算法集合。

(7) 嵌入密钥(embedding key)。嵌入标识所使用的密钥。

(8) 提取密钥(extraction key)。用于检测或提取水印的密钥。对称水印算法在嵌入和提取时需要相同的密钥。不对称水印算法使用私钥嵌入水印信息,使用公钥提取水印,同时通过公钥不能计算出私钥。

(9) 非盲方案(non-blind scheme)。是指在从测试宿主信号中提取水印时,需要提取密钥和原不含水印宿主信号的方案。

(10) 半盲方案(semi-blind-scheme)。是指在从测试宿主信号中提取水印时,需要提取密钥但不需要原不含水印宿主信号的方案。

(11) 盲方案(blind scheme)。是指在从测试宿主信号中提取水印时,既不需要提取密钥,也不需要原不含水印宿主信号的方案。

(12) I类方案(type-I-scheme)。这种方案是指检测器/提取器的输出是有效载荷或在被检测宿主信号中有意义的标识。

(13) II类方案(type-II-scheme)。这类方案需要在检测信号时被嵌入水印的知识,这

种方案只给出是否存在水印的结论。

## 2.4 本章小结

- 给出了信息隐藏的基本原理，并尽量将基本术语统一化。
- 详细给出了隐写术的三个基本协议。
- 根据不同的分类依据，介绍了各种载体的隐写过程。
- 介绍了数字水印系统，区分了数字水印与隐写术之间的差异。
- 给出了水印系统的特性，可以根据这些特性来判定相应的系统是否适用于具体的应用。

## 2.5 复习题

- 2.1 隐写术的基本原理是什么？
- 2.2 信息隐藏的分支都包括什么？
- 2.3 你使用过信息隐藏吗？你想象的信息隐藏应具有哪些特性？
- 2.4 请描述一下数字水印系统？在现实生活中，你知道具体实用的数字水印系统吗？
- 2.5 数字水印与隐写术间存在哪些相似之处？又有什么不同？
- 2.6 数字水印都有哪些特性？与隐写术要求相同吗？
- 2.7 通过因特网，可以搜索到数字水印和隐写术的内容，有时间自学一下。

## 信息隐藏的预处理

### 本章目标

- 理解加密预处理技术。
- 理解简单的图像信息伪装技术。
- 理解置乱和混沌。

信息隐藏的主要目标就是将隐藏在伪装载体中的秘密信息进行传输，并且不知情的第三方不能察觉隐藏信息的存在。在信息隐藏的不可感知性、容量及鲁棒性这三个主要特性中，隐写术更加强调不可感知性和容量，而数字水印更强调鲁棒性。在隐写术中，不可感知性占了首位，并且要求能嵌入的秘密的容量很大，三者之间需要找到最佳平衡点；信息隐藏通常都依赖于嵌入载体类型。我们所使用的载体都是以图像为载体。在信息隐藏技术的应用过程中，如果只是利用各种信息隐藏算法对秘密信息进行隐藏保密，那么攻击者只要直接利用现有的各种信息提取算法对被截获信息进行穷举运算，就很有可能提取出秘密信息。但如果我们在信息隐藏之前，先对秘密信息按照一定的运算规则进行处理，使其失去本身原有的面目，然后再将其隐藏到载体信息里面，这样所要传输的秘密信息就更安全了。即使攻击者将秘密信息从载体中提取了出来，也无法分辨出经过预处理后的秘密信息到底隐藏着什么内容，于是就认为提取/检测算法错误或该载体中不含有任何其他信息。所以，对秘密信息进行预处理是很有必要的。这也是将来我们信息隐藏技术研究的一个重要方向。也就是说，在预处理阶段，要采取各种方案对秘密信息及载体进行处理，使隐藏的信息达到第一层安全。

### 3.1 加密的预处理

在预处理部分，必须借助密码学的编码方法来实现秘密信息的预处理。在密码体制中，有许多经典算法。在信息隐藏的处理中，可选择比较适合的RC4流密码、置乱方案和混沌序列方案等。在介绍流密码之前，先介绍伪随机数发生器。

#### 3.1.1 伪随机数发生器

计算机所使用的随机数是一种伪随机数，伪随机数有多种生成算法，真正的随机数符合正态分布且其生成不能重现。但如果一个随机数能够被重现的几率很小，由此认为其为伪

随机数。伪随机数都是使用确定性的算法计算出来的,它的随机性可以用它的统计特性来衡量,其主要特征是每个数出现的可能性和它出现时与数序中其他数的关系。因为是伪随机数,所以知道了随机数算法和种子,总能够知道随机序列中任何一个随机数的值。一般来说种子的设置都是使用当前时间的毫秒数,保证随机数列的不重现性。对于伪随机数发生器的定义为:它的输出序列和真正的随机数发生器通过多项式次数试验得到的输出序列不可区分,并且任何单通路函数的发现都可以成为伪随机数发生器。一些随机函数是周期性的,虽然一般来说使用非周期性的函数要好得多,但周期性的随机函数往往快得多。有些周期函数的系数可以调整,之后它们的周期非常大,基本上与非周期的函数效果一样。也有些函数是有止尽的,用它们无法计算出无限多个伪随机数。

到现在为止,产生伪随机数最广泛使用的方法是由 Lehmer 首先提出的算法,即线性拟合法。算法有以下 4 个参数:

$m$	模	$m > 0$
$a$	乘数	$0 < a < m$
$c$	增量	$0 \leq a < m$
$X_0$	初始值或种子	$0 \leq X_0 < m$

随机数序列  $\{X_n\}$  按下面的迭代方程获得:

$$X_{n+1} = (aX_n + c) \bmod m$$

若  $m, a, c$  和  $X_0$  都是整数,那么这种方法将产生一个随机数序列,且每个随机数都满足  $0 \leq X_n < m$ 。

设计随机数发生器有三个标准:

- (1) 生成函数应是全周期的,即重复周期与  $m$  相等,也就是说, $0 \sim m$  之间的所有数都可能。
- (2) 产生的序列应显得随机,因为是采用确定性生成随机数的方法,所以是伪随机,但是有多种统计测试方法可以评估其随机程序。
- (3) 生成函数可以用 32 位运算器方便地实现。

选择合适的  $m, a, c$  可以同时满足这三点。

对于条件 1,可以证明若  $m$  是素数且  $c=0$ ,则  $a$  的某些取值可以使产生函数的周期为  $m-1$ ,只是不能得到 0 这个数。对于 32 位算术运算,  $2^{31}-1$  就是一个常用的素数,这时的产生函数为:

$$X_{n+1} = (aX_n) \bmod (2^{31} - 1)$$

$a$  的可能取值超过 20 亿个,但满足上述条件的只有其中一部分。 $a$  取值为 75,  $X_{n+1}=16807$  时,可以满足上述条件,用这个参数做成的伪随机数发生器经过了细致的测试,适用于统计和仿真。若乘数和模选择恰当,用线性拟合算法产生的随机数序列的统计特性几乎与真随机数相当。用密码编码学方法可以生成随机数。例如,循环加密,DES 输出反馈模式,另外常用的是 BBS 发生器,它是一种安全的伪随机数发生器,BBS 是 Blum、Blum、Shub 三个设计者名字的缩写,它产生的原理如下:首先,选择 2 个大素数  $p$  和  $q$ ,且要求:

$$p \equiv q \equiv 3 \pmod{4}$$

例如,  $7 \equiv 11 \equiv 3 \pmod{4}$ ,且 7 和 11 是素数,  $n=p * q$ 。

接着,选择一个随机种子数  $S$ ,要求  $S$  与  $n$  互素,即  $S$  与  $p$  或  $q$  皆无公因子。然后 BBS 按下列算法产生随机数序列:

```
X0 = S2 mod n
For i = 1 to ∞
Xi = (Xi-1)2 mod n
Bi = Xi mod 2
```

以  $B_i$  作为随机数的 1 位。如果  $n=192649=383\times503$ ,种子  $S=101355$ 。那么 BBS 生成的种子数如表 3.1 所示。

表 3.1 BBS 生成的种子数

$S$	$X_i$	$B_i$	$S$	$X_i$	$B_i$
0	20749		6	80649	1
1	143135	1	7	45663	1
2	177671	1	8	69442	0
3	97048	0	9	186894	0
4	89992	0	10	177046	0
5	174051	1			

BBS 是密码安全伪随机数发生器,它能经受连续位测试,BBS 的安全性基于对  $n$  的因子分解的困难性上。即给定  $n$ ,我们不能确定它的素因子  $p$  和  $q$ 。

### 3.1.2 RC4 流密码

本节讨论对称流密码 RC4,流密码每次加密一个字节的明文,当然流密码也可被设计为每次操作一比特或者大于一个字节的单元。典型的流密码结构如图 3.1 所示。

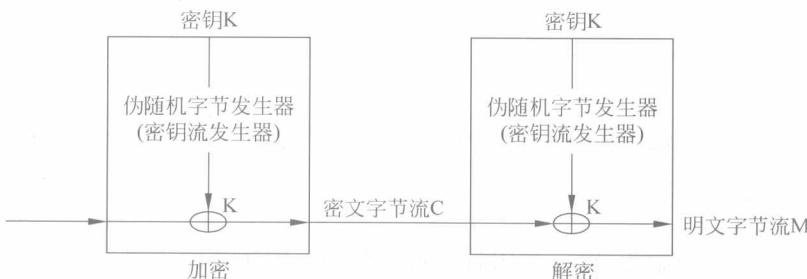


图 3.1 流密码结构图

在该结构中,密钥输入到一个伪随机数发生器,该伪随机数发生器产生一串随机的 8 位,发生器的输出称为密钥流,通过与同一时刻一个字节的明文流进行异或操作产生密文流。如果发生器产生的下一个字节是 01101100,而下一明文字节为 11001100,则得出密文字节为: 10100000。解密时需要使用相同的随机序列,即 10100000 与 01101100 异或得出明文 11001100。

在设计流密码时要考虑:

- (1) 加密序列的周期要长。
- (2) 密钥流应尽可能接近于真正的随机数流的特征。
- (3) 在流密码结构中随机数发生器的输出取决于输入密钥的值,应该保证密钥长度不小于 128 位。

下面我们详细介绍 RC4 算法。

RC 是 Ron Rivest 设计的一种流密码,它是一个可变密钥长度,面向字节操作的流密码。该算法以随机置换为基础,RC4 是最广泛使用的流密码,它的算法描述如下:用从 1~256 个字节的可变长度密钥初始化一个 256 个字节的状态矢量 S,S 的元素记为  $S[0]$ , $S[1], \dots, S[255]$ ,从始至终转换后的 S 包含从 0~255 的所有 8 位。对于加密和解密,字节 k 由 S 中 255 个元素按一不定期方式选出一个元素而生成。每生成一个 k 值,S 中的元素被重新转换一次。

- 初始化 S

开始时,S 中元素的值被置为按升序从 0~255,同时建立一个临时矢量 T。如果密钥 K 的长度为 256 字节,则将 K 赋给 T。否则,若密钥长度为 keylen 字节,则将 K 的值赋给 T 的前 keylen 个元素,并循环重复用 K 的值赋给 T 剩下的元素,直到 T 的所有元素都被赋值。预操作可概括如下:

```
/* Initialization */
For i = 0 to 255 do
  S[i] = i;
  T[i] = k[i mod keylen];
```

然后用 T 产生 S 的初始置换。从  $S[0] \sim S[255]$ ,对每个  $S[i]$ ,根据由  $T[i]$  确定的方案,将  $S[i]$  置换为 S 中的另一字节:

```
/* Initial Permutation of S */
j = 0;
For i = 0 to 255 do
  j = (j + S[i] + T[i]) mod 256;
  swap(S[i], S[j]);
```

因为对 S 的操作仅是交换,所以唯一的改变就是置换。S 仍然包含所有值为 0~255 的元素。

- 密钥流的生成

矢量 S 一旦完成初始化,输入密钥就不再被使用。密钥流的生成是从  $S[0]$  到  $S[255]$ ,对于每个  $S[i]$ ,根据当前 S 的值,将  $S[i]$  与 S 的另一字节置换。当  $S[255]$  完成置换后,操作继续重复,从  $S[0]$  开始:

```
/* Stream Generation */
i, j = 0;
while (true)
  i = (i + 1) mod 256
  j = (j + S[i]) mod 256;
  swap(S[i], S[j]);
  t = (S[i] + S[j]) mod 256;
  k = S[t];
```

加密中,将 k 的值与下一明文字节异或;解密中,将 k 的值与下一密文字节异或。

通过 RC4 算法,可以生成安全的加密密钥,这样就在预处理层中实现了秘密消息的安全性。

我们要嵌入的信息是文本,可以通过上述算法对其加密。

## 3.2 简单的图像信息伪装技术

另外如果要嵌入的对象是图像,则对图像的预处理的方式很多,这里介绍简单的图像伪装技术与置乱方法。

这里我们分别以 .bmp 和 .gif 格式的图像为载体。对载体进行伪装。BMP 文件包括各种信息的头记录和带着像素数据的扫描线。对于此类文件,可以对像素数据扫描线加密,也可以对调色板加密来进行信息伪装。我们使用 Cryptbmp. exe 对图像进行加密,如图 3.2 所示。

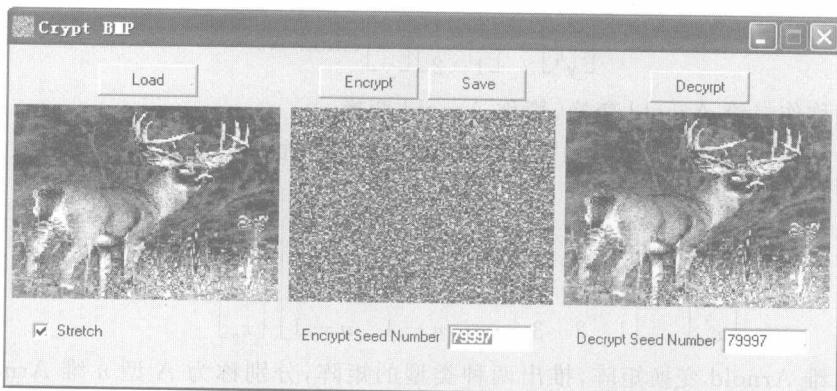


图 3.2 简单的图像加密

在加密的过程中,种子数的选择很重要。如果我们假设 R 是随机的比特流顺序,A XOR B 的结果与(A XOR R) XOR (B XOR R) 结果相同,但还原时得到的图像是不同的。所以对于不同图像一定要使用不同的唯一密钥。

对于 JPEG 格式的文件,需要将加密的信息作为二进制信息流进行保存。JPEG 加密技术可以应用于任何其他类型的图像文件,比如 GIF。加密过的 JPEG 文件是一串二进制比特流,只有在经过解密以后才能使用。下面我们使用 crypjpeg. exe 对 JPEG 类型的文件进行简单的预处理,形成二进制流。用一串随机产生的字符串异或的方法来加密是一种简单加密的方法。这样的方法也对其他类型的文件有效。

## 3.3 置 乱

信息隐藏技术可以将机密的图像、语音或文字等信息进行置乱加密,然后隐藏在可公开的载体图像中,这样使别人无法察觉秘密的存在,从而实现了隐写,现有的图像置乱加密技

术有：Arnold 变换、Hilbert 曲线变换、Fibonacci 变换、幻方变换、FASS 曲线、分形技术、幻方、正交拉丁方、骑士巡游、仿射变换、原根、Gray 码变换的置乱方法等。数字图像的置乱就是一种可逆变换，它是在二维的层次上，对数字图像色彩、位置、频率进行干扰来扰乱图像，使置乱后的图像杂乱无章，如果不知道置乱的类型，很难恢复出图像。数字图像置乱可以在数字图像的空间域，如颜色空间、位置空间上进行，也可以在数字图像的频域上进行。数字图像置乱可分为：基于循环移位的数字图像置乱，基于异或操作的数字图像置乱，基于幻方的数字图像置乱，基于空间填充曲线的数字图像置乱，如弓形线、zigzag、螺旋曲线等，基于 Arnold 变换的数字图像置乱等。下面分析和实现典型的置乱。

### 1. Arnold 置乱

首先分析基本 Arnold 变换。

**定义 1** 设有单位正方形上的点  $(x, y)$ ，将点  $(x, y)$  变成另一点  $(x', y')$  的变换，如式(3-1)所示：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod{N} \quad (3-1)$$

此变换称作二维 Arnold 变换，简称 Arnold 变换。

**定义 2** 基本 Arnold 变换推广到  $n$  维 Arnold 变换如公式(3-2)所示：

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 2 & & 1 & \\ & & & \ddots & & \\ & & & & n-1 & n-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \pmod{N} \quad (3-2)$$

根据  $n$  维 Arnold 变换矩阵，推出两种类型的矩阵，分别称为 A 型  $n$  维 Arnold 变换矩阵和 B 型  $n$  维 Arnold 变换矩阵。

#### (1) A 型 Arnold 变换矩阵

massey 在其设计的 SAFER 类型分组密码中使用了一种“伪随机哈达马达变换 (pseduo-Hadamard transform)”，简称 PHT，其中 2-PHT 定义如公式(3-3)所示：

$$H_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \quad (3-3)$$

并把二维 PHT 推广到  $n$  维，记为  $n$ -PHT，如公式(3-4)所示：

$$H_n = \begin{bmatrix} 2 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 1 & \cdots & 1 & 1 \\ & & & \ddots & & \\ 1 & 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \quad (3-4)$$

为了和二维 Arnold 变换矩阵相一致，同时受  $n$ -PHT 启发，定义了如下的  $n$  维 Arnold 变换矩阵。

**定义 3** A 型  $n$  维 Arnold 变换矩阵为如公式(3-5)所示  $n \times n$  矩阵：

$$Ra_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & 1 & \cdots & 1 & 2 \end{bmatrix} \quad (3-5)$$

A型  $n$  维 Arnold 变换矩阵有如下两个性质：

- ① A型 Arnold 变换矩阵的行列式恒为 1。
- ② A型  $n$  维 Arnold 变换矩阵的可逆矩阵如公式(3-6)所示：

$$Ra_n = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \vdots & & \vdots & \\ -1 & 0 & 0 & \cdots & 1 & -1 \\ n-1 & -1 & -1 & \cdots & 1 & 1 \end{bmatrix} \quad (3-6)$$

(2) B型 Arnold 变换矩阵

**定义 4** 对任意正整数  $N$ , B型  $n$  维 Arnold 变换矩阵为以下  $n \times n$  矩阵, 如公式(3-7)所示:

$$R(b)_n = \begin{bmatrix} b & b & b & \cdots & b & b \\ b & b+1 & b+1 & \cdots & b+1 & b+1 \\ & & & \vdots & & \\ b & b+1 & b+2 & \cdots & b+(n-2) & b+(n-2) \\ b & b+1 & b+2 & \cdots & b+(n-2) & b+(n-1) \end{bmatrix} \quad (3-7)$$

其中  $b$  为 NZ 上的可逆元。

从定义可看出, 公式(3-6)中定义的  $n$  维 Arnold 变换矩阵(即  $n$  维 Arnold 变换矩阵)是  $b=1$  时的 B型  $n$  维 Arnold 变换矩阵。

下面分析和改进 Arnold 置乱加密算法。

设原图像  $A$  为  $M \times N$  大小, 假定像素位置为  $(x, y)$ , 按几何变换置乱后的像素位置为  $(X, Y')$ , 则几何变换的置乱方法如下:

$$[x', y'] = [a, b, c, d] * [x, y] (\bmod N) \quad (3-8)$$

其中, 要求  $ad - bc = 1, a, b, c, d \in \mathbb{Z}$  ( $\mathbb{Z}$  是正整数集合), 其解有很多, 当  $a = b = c = 1, d = 2$  时, 就是著名的 Arnold 变换, 它是数学家 Arnold 在研究遍历理论时提出的一种变换, 俗称猫脸变换, 原意为 cat mapping, 设想在平面单位正方形内绘制一个猫脸图像, 通过此变换, 猫脸图像将由清晰变模糊, 这就是 Arnold 变换。Arnold 变换实际上是一种点的位置移动, 且这种变换是一一对应的。此外, 这种变换可以迭代地做下去。类似的变换还有面包师变换。Arnold 变换具有周期性, 即当迭代到某一步时, 将重新得到原始图像。Dyson 和 Falk 分析了离散 Arnold 变换的周期性, 给出了 Arnold 变换的周期。

对于数字图像来说, 可以将其看成是一个函数在离散网格点处的采样值, 这样就得到了一个表示图像的矩阵。矩阵中元素的值是对应点处的灰度值或 RGB 颜色分量值。位置的移动实际上是对应点的灰度值或者 RGB 颜色值的移动, 即将原来位置点像素对应的灰度值

或 RGB 颜色值移动至变换后的位置点。如果对一个数字图像迭代地使用离散化的 Arnold 变换，即将左端输出的作为下一次 Arnold 变换的输入，可以重复这个过程一直做下去，当迭代到某一步时，如果出现的图像符合对图像的“杂乱无章”标准的要求，这即是一幅置乱了的图像。

例：置乱的 Matlab 的实现

```
% 嵌入图像并显示
w = imread('arnold.bmp');
M = w;
Size_w = size(w);
subplot(2,2,2);
imshow(w);
title('将要嵌入的图像');

% 取水印图像的维数
[c,d,e] = size(w);

% 定义一个零矩阵用于设置 Arnold 变换后生成的新图像
w1 = zeros(c,d,e);

% 读取灰度值矩阵 U
U = w(:,:,:,1);

% 读取灰度值矩阵 U
U1 = w1(:,:,:,1);

% 逐行扫描水印图像的坐标 x,y 所构成的矩阵
for i = 1:c
    for j = 1:d
        % 对每个像素的 x,y 坐标进行 Arnold 变换
        i1 = i + j;
        j1 = i + 2 * j;
        i1 = mod(i1,1);
        j1 = mod(j,1);

        % 将每个像素的 x,y 坐标对应的灰度值放入定义的数组 E 中
        if ((i1~=0)&(j1~=0))
            U1(i1,j1) = U(i,j);
            w1(i1,j1) = w(i,j);
        end
    end
end

subplot(2,2,3);
imshow(w1);
title('置乱图像');
```



图 3.3 置乱处理后的图像

仿真实验结果如图 3.3 所示。

实验结果表明，经过置乱处理后的图像，已经看不出原图像，这就达到了预处理的目的。

例：改进算法

```
A = imread('DEER.BMP');
N = length(A);
% 按列抽样
T = zeros(N);
for i = 1:N/2
```

```

T(2 * i - 1, i) = 1; % 第一个元素成对称
End
for j = 1: N/2
    T(2 * j, j + N/2) = 1;
End
B = double(A) * T;
% 按行抽样
T = zeros(N);
for i = 1: N/2
    T(i, 2 * i - 1) = 1;
End
for j = 1: N/2
    T(j + N/2, j) = 1;
End
B = T * B;
imshow(uint8(B)) % 显示置乱后的图像 B

```

仿真的实验结果如图 3.4 所示。这能达到更佳的预处理的目标。

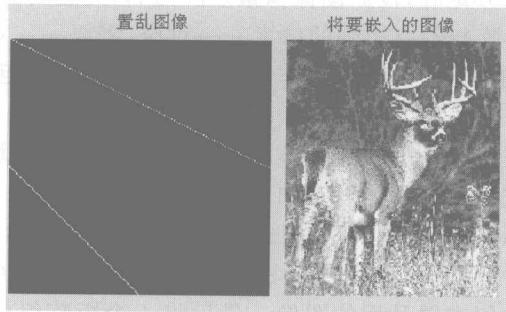


图 3.4 Arnold 算法的置乱

另外,通过前面的矩阵分析,可以知道,迭代次数的选取是不同的,出现的结果也就不同,在迭代次数不断改变的情况下,取相应的迭代系统数,就能在提取图像时恢复原图像。另外,还通过进一步的仿真实验来改进上述的 Arnold 置乱。

Arnold 置乱有优势也有缺点,在选择图像时,必须是方阵图像,否则就不能使用 Arnold 置乱进行预处理。为了解决这个问题,下面将讨论可以在预处理中使用的其他置乱方法。

## 2. 幻方置乱

幻方矩阵就是以自然数  $1, 2, \dots, n$  为元素的  $n$  阶矩阵,如公式(3-9)所示:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad (3-9)$$

有了幻方矩阵的定义后,接下来看如何构造幻方矩阵。幻方变换根据矩阵中自然数序号来对图像块位置进行移动。如果  $n$  阶图像相对应的  $n$  阶方阵为  $B$ ,幻方矩阵为  $A$ ,变换过程是首先将  $A$  中自然序号与  $B$  中像素点一一对应,然后将  $A$  中的序号为  $m$  的元素移动到

$m+1$  的位置,将序号为  $n^2$  的元素移到序号为 1 的位置。 $A$  中元素  $m$  位置的像素移至元素  $m+1$  位置处,1 变为原来 2 的位置,2 变为原来 3 的位置,15 变为原来 16 的位置,16 变为原来 1 的位置,随着位置的移动,到一定位置,一定能变幻出原有的图像。也就是说当达到  $n^2$  次变换时,就可以恢复出图像。利用  $B=\text{magic}(4)$  可以构造出第一个幻方矩阵,如公式(3-10)所示。

$$A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix} \rightarrow B = \begin{bmatrix} 15 & 1 & 2 & 13 \\ 4 & 10 & 9 & 7 \\ 8 & 6 & 5 & 11 \\ 3 & 13 & 14 & 16 \end{bmatrix} \quad (3-10)$$

式中, $B$  即第一次幻方所得的矩阵。

也就是说,基于幻方的变换具有周期性,对于一幅像素为  $n \times n$  的数字图像,其变换周期为  $n^2$ 。通过仿真实验可知,基于奇数阶幻方变换的数字图像  $P_{n \times n}$  经  $k^n$  次迭代变换,图像恢复, $k^n$  为变换的准周期, $k=1, 2, \dots, n-1$ 。准周期性形成的原因因为奇数阶幻方的构造算法和幻方的构造算法共同所至,每变换  $k^n$  次时,图像像素总体在行位置上共计下移  $2^k$  行,列位置上共计左移  $k$  列,使图像恢复。

首先需要生成  $n$  阶幻方矩阵,设原图像像素矩阵为  $A$ ,经过幻方置乱后的图像像素矩阵为  $B$ 。因为要将原像素矩阵与幻方矩阵的像素一一对应,需要一一对应赋值,当位置第一次改变时,得到第一次幻方变换,输出的矩阵作为下一次幻方的输入矩阵,经过  $n$  次处理后,得到需要的置乱矩阵。

算法的结果如图 3.5 所示。另外,还可以将幻方变换与图像分块相结合,这样就能达到更好的置乱效果。此改进算法的具体思路如下:

源图像像素矩阵为  $A$ ,按上述的算法对图像进行一次置乱。然后对第一次置乱后的图像进行分割,再根据第一次置乱的算法对各图像分块进行幻方置乱(一般可以分成 4 块等),这样就得到了 4 个又一次幻方置乱后的图像。最后将 4 个矩阵组成图像的像素矩阵,这样递归的幻方下去,即可得到理想的幻方置乱矩阵。

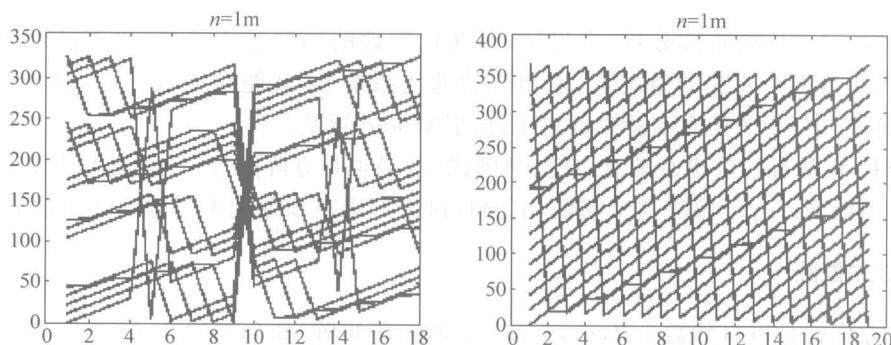


图 3.5 幻方变换

### 3.4 混沌

混沌是指发生在确定性系统中的貌似随机的不规则运动,一个确定性理论描述的系统,其行为却表现为不确定性、不可重复和不可预测,这就是混沌现象。进一步研究表明,混沌

是非线性动力系统的固有特性,是非线性系统普遍存在的现象。牛顿确定性理论能够完美处理的多为线性系统,而线性系统大多是由非线性系统简化来的。混沌系统具有优良的特性:构造简单、对初始值的敏感性和类白噪声。利用初值可以精确地重构混沌序列。那么我们可以借助混沌的特性,利用 Logistic 混沌来进行预处理,进一步增强破解难度,提高秘密信息的安全性。

Logistic 混沌序列的遍历统计特性近似于零均值白噪声,具有良好的随机性、相关性和复杂性,不可能对混沌序列进行正确的长期预测。Logistic 混沌序列定义为:

$$x_{k+1} = \mu x_k (1 - x_k) \quad (3-11)$$

其中,  $0 < \mu \leq 4$  称为分支参数。非发散动力系统中的 Lyapunov 指数用来判别系统的混沌判别,通常混沌系统具有正的 Lyapunov 指数。当  $\mu$  限制在  $3.57 \leq \mu < 4$  的狭窄范围内时, Lyapunov 指数大于 0。此时系统呈现混沌状态。因而实际应用中,分支参数  $\mu$  限制在区间  $[3.57, 4]$  内。

尽管在理论上是非周期的,但由于计算机在计算时是离散的,实际上,计算的离散性使得混沌序列出现周期性重复现象而失去混沌特性。为了避免这种现象,实际生成混沌序列时,需要对  $x_k$  进行微小的反馈扰动。我们采用的方法是利用前一个点对现在的取值进行微调扰动。伪码程序为:

```
if (mod(i, 4) == 0)
    p(i) = 0.999 * (u * p(i - 1) * (1 - p(i - 1)) + 0.001 * p(i - 1));
else
    p(i) = 1.001 * (u * p(i - 1) * (1 - p(i - 1)) - 0.001 * p(i - 1));
end;
```

图 3.6 是经过扰动处理后 Logistic 混沌序列图。从图中可以看出,利用混沌序列生成的密钥流具有良好的随机统计特性和均匀分布特性。

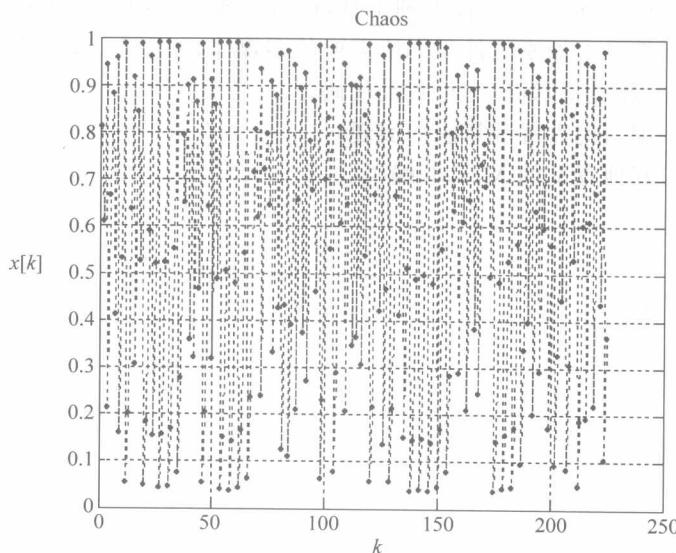


图 3.6 Logistic 混沌序列

### 3.5 本章小结

在信息安全方面,为了最大程度地使要传输的秘密信息不被恶意攻击者发现和提取,就需要对秘密信息进行预处理,这样使秘密信息达到第一层安全。本章分析和改进了随机数发生器算法和 RC4 算法的预处理以及置乱和幻方的预处理。在使用置乱算法时,不应只简单地使用一种置乱算法,而应同时使用几种算法,使攻击者即便用穷举算法,也无法提取出想要的秘密信息。

- 总结了隐写模型,并将典型的模型进行改进。
- 详细分析、改进和实现了基于文本和图像的预处理算法。
- 对典型的预处理算法进行了改进,并做了大量仿真实验。
- 对 A 型  $n$  维 Arnold 变换矩阵进一步推广,进一步扩大了图像隐藏的加密信息容量。

另外,值得进一步研究的问题还有:

- 进一步改进预处理算法的实用性和相关等特性。
- 对各预处理算法进行评价,提出相应的评价方案,使预处理阶段有章可循,即建立完善的理论体系。

### 3.6 复习题

- 3.1 为什么要在信息隐藏之前进行预处理?
- 3.2 请编写程序生成随机种子数。
- 3.3 试编写流密码程序,语言不限。
- 3.4 找到 encryptbme.exe 图像处理工具,试着完成简单图像的加密。
- 3.5 使用你所熟悉的加密软件为你所需要的多媒体,如文本、音频或图像加密。
- 3.6 在 Matlab 环境下,调试运行置乱代码。
- 3.7 自己动手编写幻方置乱算法的实现。
- 3.8 尝试编写其他置乱的 Matlab 编码。

## 信息隐藏模型

### 本章目标

- 理解隐写术基本模型。
- 了解隐写术的安全模型。
- 理解隐写术模型所占有的理论意义。
- 理解数字水印模型。
- 理解感知模型。

信息隐藏模型的研究是信息隐藏基本理论、基本原理研究的基础。在本章中针对信息隐藏的两个重要应用：隐写术与数字水印，我们分析相关的模型以及模型的评价及相关分析。从广义上讲，模型分为两类，一类是从通信角度来分析的通信模型；另一类是从空间角度分析算法。

### 4.1 隐写术模型分析

隐写术主要是将秘密信息隐藏，然后以隐蔽通信的方式不让未授权第三方察觉，所以我们首先从通信的角度来分析隐写术模型，然后从几何的空间角度来分析数字水印模型。

#### 4.1.1 Simmons 模型分析

在隐写术的相关文献中，普遍采用 Simmons 提出的模型和模型的变种，隐写术的理论模型是研究隐写术的基础。隐写术早期的研究是将秘密信息直接加到载体信息中，将载体数据看作噪声，若解码者知道噪声，则提取时用伪装载体数据减去噪声即得秘密信息。目前，隐写术是一种隐蔽通信模型。对隐写术系统的模型理论、检测理论和容量理论都需要进一步研究，这就形成了隐写术的理论研究。在隐藏容量研究方面，通信模型为： $Y_n = X_n + S_n + Z_n$ ， $X_n$  为隐藏的信息， $S_n$  为载体， $Z_n$  为攻击噪声，可以得出隐藏容量  $C = 1/2\log(1 + P/N)$ ， $P$  是  $X$  的平均能量， $N$  是  $Z$  的平均能量。

在 1983 年，Simmons 针对隐蔽通信提出了第一个隐写术的场景描述：囚犯问题。假定 Alice 和 Bob 是分别处在不同牢房中的囚犯，为了合谋一次越狱行动，相互间需要秘密通

信,而他们的每一次通信都必须经过看守人 Wendy 的监督。Wendy 可以阅读所有囚犯的信件,并决定是否传送或不传送这些信件,同时还可以对信件进行修改,但并不改变信件的内容。为了使通信不被怀疑,Alice 和 Bob 不能采用常用的密码通信技术,因为一封经过加密且语义混乱的密信虽然可能不会泄露计划,但已经足以作为两个犯人图谋不轨的证据。因此,Alice 和 Bob 不仅要保证密信不可破解,而且要隐藏秘密通信的事实。囚犯问题如图 4.1 所示。

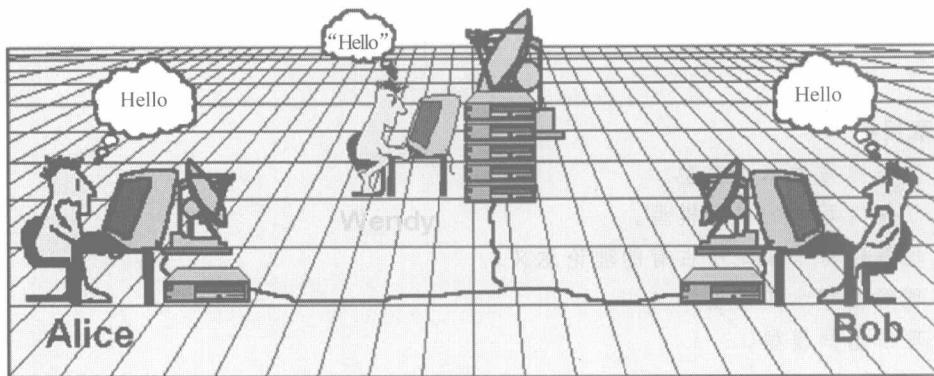


图 4.1 囚犯问题

这就是典型的隐写术问题的图例。

这个模型强调的是如何保证隐蔽通信的实现,并没有提出实现隐写术的原理,所以实际上,这个模型是隐蔽通信的模型。我们在第 5 章中将研究隐蔽通信。

通常称需要隐藏的信息为秘密信息,而公开的已嵌入秘密信息的称为伪装载体,隐写的过程一般由隐写密钥来控制,通过嵌入算法将秘密隐写术于公开信息中,而伪装载体则通过信道传递,通信对方用检测器或利用密钥从伪装载体中恢复/检测出秘密信息。秘密信息可以是版权信息、秘密数据或者序列号等;而公开信息则称为载体信息,如视频、音频片段。这种隐写术过程一般由密钥(key)来控制,即通过嵌入算法(embedding algorithm)将秘密隐写术嵌于公开信息中,而隐蔽载体(隐藏有秘密信息的公开信息)则通过信道(communication channel)传递,然后检测器(detector)利用密钥从隐蔽载体中恢复/检测出秘密信息。从上面分析可知,隐写术技术主要由两部分组成:信息嵌入算法,它利用密钥来实现秘密信息的隐藏。隐蔽信息检测/提取算法(检测器),它利用密钥从隐蔽载体中检测/恢复出秘密信息。在密钥未知的前提下,第三者很难从隐秘载体中得到或删除,甚至发现秘密信息。

#### 4.1.2 通信系统模型分析

在本小节中,重点从信息论的观点,揭示信息隐藏内在的性质,应用信息论描述信息隐藏的嵌入,提取和检测模型,从而为研究信息隐藏的容量分析,极限隐藏,以及指导设计隐藏算法提供坚实的基础,建立其信息论观点的理论模型。先分析通信基本通信模型,然后提出隐写术通信模型,从而使隐写术的安全达到一个新的理论高度。

下面首先介绍通信系统模型。

图 4.2 给出了一个传统通信系统的基本结构。M 是准备发送的信息，信道编码器对信息 M 进行编码，准备发送，它将所有可能的信息映射为码字 X，X 可以在信道中传输的符号所组成的集合中选择得到。码字序列通常标记为 X。噪声 N 与 X 形成 Y，Y 到达信息解码器后被解码成信息  $M_n$ ，标准的通信模型如图 4.2 所示。

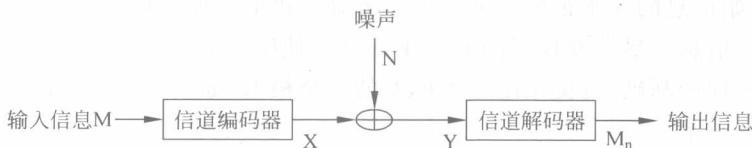


图 4.2 通信系统的标准模型

为了进一步保证传输信息的安全，可以将传输的密码技术加入，基本通信系统图变成基于密钥的信道编码的通信信道基本模型，模型如图 4.3 所示。

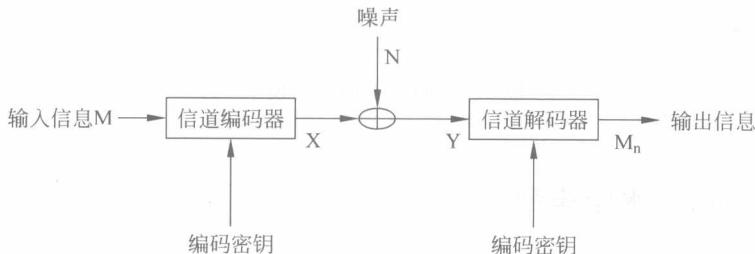


图 4.3 基于密钥的信道编码的通信信道模型

而在隐写术的通信中也应考虑像 Wendy 这样的主动攻击的存在，所以基本的隐写术模型如图 4.4 所示。

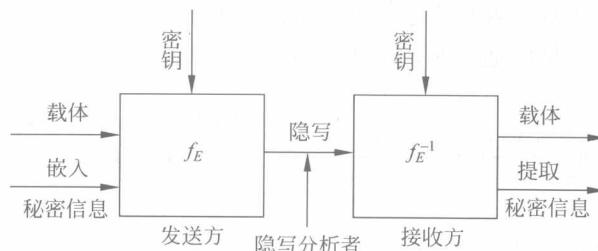


图 4.4 基本的隐写术理论模型

上图中所讨论的系统都符合 Kerchhoff's 准则，即加密函数、解密函数以及隐藏信息嵌入函数  $f_E$  和提取函数  $f_E^{-1}$  是公开的。

### 4.1.3 隐写术的安全模型分析

对于隐写术而言，多层安全机制会让秘密信息更加安全。在隐写术中加密与嵌入算法以及隐蔽传输都很重要。保持密钥的算法是很困难的。因为优秀的加密系统遵循 Kerchhoff's 原则，算法安全取决于密钥，如果没有安全密钥，能攻破算法的几率很低。因此，

许多加密工具都使用公开加密算法。如果文本信息正在通过不安全的信道，并且文本信息已经通过了隐写处理，通信双方并不会怀疑正在进行秘密通信。另外，如果对文本进行多层次安全处理，那么文本信息的安全性将更高。对加密过的信息再通过隐写工具进行处理，那么将为秘密再加一层安全。如果隐写分析者检测到信息的存在，他需要提取原始信息，就如前面所述，提取原始信息的几率很低。如果隐写者确实提取了原始信息，但是他还需要密钥以解密来得到原始信息。尽管发现了隐藏的秘密已经使隐写失败，但这仍能保护信息不被破解。根据前面的理论基础，可提出隐写术模型的安全模型，如图 4.5 所示。

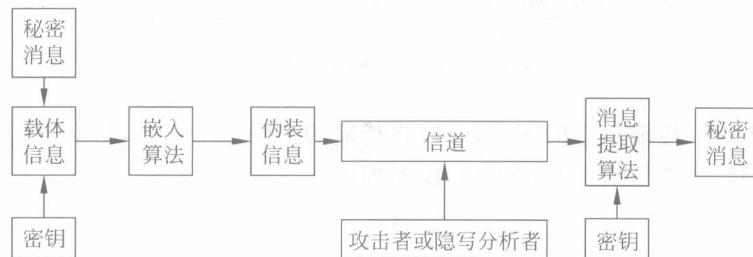


图 4.5 隐写术的安全模型

#### 4.1.4 基于通信的水印模型

Cox 于 1995 年在其著作中给出了基于通信的基本模型，认为嵌入编码器到检测编码器之间的信息传输可以看做是一种通信，并且需要将信息映射为同载体维度相同、类型相同的模板。在 Costa 的脏纸通信的研究成果基础上，Gelfand 和 Pinsker 结合 Cox 的边信息水印通信模型，形成了信息隐藏的信息论模型，将信息隐藏看做是隐藏者和攻击者之间的通信博弈，其中的隐藏者和合法解码者拥有边信息，给出了具有边信息信道的容量定义。

如图 4.6 所示，是最早的通信水印模型。如图 4.7 所示，是数字水印的边信息模型。

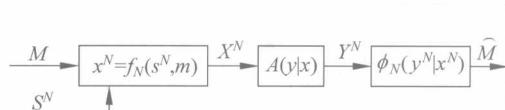


图 4.6 早期的数字水印模型

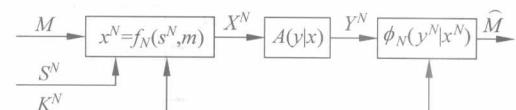


图 4.7 数字水印的边信息通信模型

## 4.2 数字水印空间模型

根据信息隐藏活动的离散化特点，除了可以将数字水印看做是传统的通信模型之外，最常见的还是几何模型。几何模型有时也称为空间模型。

空间模型对信息隐藏的对象和过程进行细化和抽象，根据信息隐藏活动的离散化特征，利用集合论和矩阵空间概念描述信息隐藏的基本问题。模型对信息隐藏各元素进行空间概念表示，对信息隐藏过程进行空间转换描述。

下面介绍图 4.8 所示数字水印空间模型所用到的空间。

(1) 信息空间(information space)。信息存在形式的空间表示。信息空间中的元素有特定的结构和表示。同一信息可以存在于不同的信息空间中。

(2) 空间变换(space transform)。不同信息空间之间的点有一定的变换关系。我们经常使用各类空间变换,如空域和变换域,这种变换只是改变信息空间中信息的表示,而信息本身并不发生任何变化。

(3) 载体空间(carrier space)。载体所存在的信息空间。

(4) 宿主空间(host space)。宿主信息所存在的空间。

(5) 隐藏空间(secret space)。宿主空间中真正用于隐藏数据的子空间。

(6) 秘密消息空间(message space)。原始隐藏数据所存在的信息空间。

(7) 寄生空间(hermit space)。为了保证秘密信息的安全,在秘密信息嵌入之前需经预处理,然后再嵌入隐藏空间,预处理之后的数据所存在的信息空间称之为寄生空间。

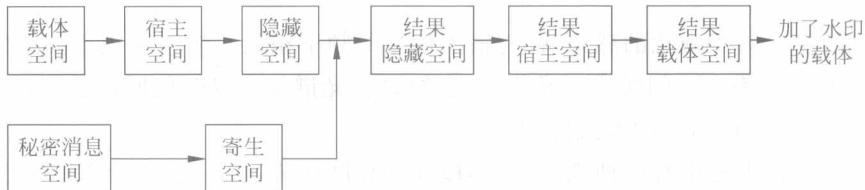


图 4.8 数字水印空间模型

## 4.3 感知模型

不可感知性是信息隐藏的很重要的衡量指标。到目前,根据人类听觉和视觉系统,已经提出了许多感知模型,因为人类感知存在很大个体差异,本节首先对人类感知进行分析,然后利用人类视觉感知特点和矩阵理论来描述信息隐藏的基本问题,主要介绍 Watson 的感知模型。

### 4.3.1 人类感知

人眼对在波长范围为  $400 \sim 770\text{nm}$  的电磁辐射非常敏感。彩色图像可以用函数  $C(x, y, t, \lambda)$  表示。这个函数能表示位置  $(x, y)$ ,反射光的波长  $\lambda$  和动态图像情况下的时间。对于颜色视觉,有三个基础可用的光谱敏感度函数  $V_R(\lambda)$   $V_G(\lambda)$  和  $V_B(\lambda)$ 。通过人眼或照相机系统对场景中对象进行感知主要是通过它的辐射  $R(\lambda, x, y, t)$ 。照明与主观人类感知之间,以及与人类响应之间都存在着直接关系。在 Weber 定律中,第一次公式化地给出了这种关系。公式表示为

$$\frac{W_L}{L} = k \quad (4-1)$$

实验调查表明,Weber 定律只适用于中间照明值,对于很高和很低的照明值都不适用。

可感知的亮度  $B$  和照明  $L$  间的关系是对数关系, 即  $B \propto \log L$ 。

根据 Thomas Young 的三原色理论, 我们视觉系统感知的所有颜色都是基本颜色的线性组合。如果两种颜色之间存在最小可觉差(just noticeable difference, JND), 实际最小可觉差值不是常数, 这主要因为人类视觉的非线性和 RGB 空间的不均匀性。如果人眼颜色感知能力还未饱和, 在较强照明时, 人眼颜色感知能力更好。Buchsbaum 非线性等式的常数考虑了眼睛的适应条件和照明条件。Buchsbaum 在 Weber 研究的基础上开始了他的研究, 分析获得了对数形式的视觉非线性。

### 4.3.2 评价的基本指标

感知模型实际上是函数  $D(C_0, C_s)$ , 这个函数是计算原图像和隐写后图像之间的感知距离, 传统使用的 MSE 均方差就是一个最简单的距离函数。感知模型的基本指标是灵敏度, 掩蔽及合并。

(1) 灵敏度。对于视觉的感知, 都与输入信号的频率相关。频率响应主要指空间频率、光谱频率和时间频率。空间频率通常指亮度敏感度, 光谱频率以颜色形式感知, 一般为低频响应, 时间频率响应以运动的形式感知。

(2) 掩蔽。在视觉中有两种掩蔽, 频率掩蔽和亮度掩蔽, 前者是一段频率对另一段频率的感知掩蔽, 后者是局部亮度会掩蔽对比度变化。

(3) 合并。在感知距离模型中, 对多个不同失真的感知性综合归一为对伪装载体的评价, 称为合并, 公式为:

$$D(C_0, C_s) = \left( \sum_i |d[i]^p| \right)^{\frac{1}{p}} \quad (4-2)$$

### 4.3.3 Watson 感知模型

Watson 提出了一个测量视觉保真度模型, 它估计了原图像和伪装图像之间的 JND 值, 对于噪声加入图像后产生的影响估计, 此模型比 MSE 效果要好。

模型的基本原理是根据图像的块离散 DCT 估计变化的感知性, 然后将这些估计合并成对感知距离的单个估计。此模型基于 DCT 域, 经过分块量化处理后, 图像能量集中在每一块的低频部分, 这样就能估计量化噪声的感知度, 就可以根据每幅图像的具体特性来改变量化步长。在信息隐藏中需要用此模型评价和控制信息隐藏的算法。

Watson 模型由一个敏感度函数、两个基于亮度和对比度掩蔽及合并部分组成。

(1) 模型定义了频率敏感度表, 表中每个元素表示每块中不存在任何掩蔽噪声时, 可感知 DCT 系数的最小幅度值。DCT 频率敏感度表如表 4.1 所示。这个表只是一个根据图像的分辨率和观察者到图像的距离, 根据公式计算 DCT 系数的最小幅度值, 即单位 JND 的系数值。当前两者变化时, 表中的值也会发生变化, 这只是一个示例。

表 4.1 DCT 频率敏感度表

1.40	1.01	1.16	1.66	2.4	4.43	4.79	6.56
1.01	1.45	1.32	1.52	2.0	2.71	4.67	4.93
1.16	1.32	2.24	2.59	2.98	4.64	4.6	5.88
1.66	1.52	2.59	4.77	4.55	5.3	4.6	7.6
2.4	2.00	2.98	4.55	6.15	7.46	6.28	10.17
4.43	2.71	4.64	5.3	7.46	9.62	8.71	14.51
4.79	4.67	4.6	6.28	8.17	11.58	11.58	17.29
6.56	4.93	5.88	7.6	10.17	14.51	14.5	21.15

(2) 亮度掩蔽,如果 DCT 块的平均亮度值较高,DCT 系数较大的修改也不易察觉,Watson 模型对每个像素块,根据敏感度表的值进行调整,亮度的掩蔽阈值  $t_L[i, j, k]$  为:

$$t_L[i, j, k] = t[i, j](C_0[0, 0, k]/C_{0,0})^{a^T} \quad (4-3)$$

(3) 上述亮度掩蔽值也受到对比度掩蔽的影响,对比度掩蔽阈值  $s[i, j, k]$  为:

$$s[i, j, k] = \max\{t_L[i, j, k], |C_0[i, j, k]|^{\omega[i, j]} t_L[i, j, k]^{1-\omega[i, j]}\} \quad (4-4)$$

(4) 合并,对原始图像与伪装图像比较,计算对应 DCT 系数的差值,然后除以对比度掩蔽,得到每项的可感知距离  $d[i, j, k]$ ,然后合并成一个总的感知距离为:

$$D_{Wat}(C_0, C_S) = \left( \sum_i |d[i, j, k]^p| \right)^{\frac{1}{p}} \quad (4-5)$$

因为此模型是基于 DCT 变换的,主要用于 JPEG,经过变换之后,可以将能量集中在少量的低频系数之上,然后通过此模型来估计量化后的感知度。

## 4.4 本章小结

信息隐藏模型在信息隐藏研究中起着举足轻重的作用,本章主要分析了信息隐藏模型,分别从通信角度和几何角度来对不同模型进行分析、描述。利用已有的集合论和矩阵空间概念对信息隐藏活动进行了形式化描述。这将有助于发现新的、更加有效的信息隐藏算法。

需要进一步做的工作是各类模型都是探索性模型,还需要进一步量化分析,这需要大量的时间和精力才能完成。

## 4.5 复习题

- 4.1 隐写术的安全模型是如何理解的?
- 4.2 Simmons 模型的主要优缺点是什么?
- 4.3 请描述典型的数字水印模型。
- 4.4 什么是感知?
- 4.5 为什么需要信息隐藏的模型? 你能建立自己的模型吗?

# 第5章

## 信息隐藏算法

### 本章目标

- 了解信息隐藏算法的分类方式和分类依据。
- 理解位平面算法。
- 理解调色板算法。
- 理解空域算法。
- 理解频域算法。
- 理解小波变换算法。
- 了解统计方法和图像融合算法。

根据相应的信息隐藏模型,在完成秘密信息预处理之后,就需要按照某种嵌入算法将处理后的信息嵌入到载体中,信息隐藏算法的研究在信息隐藏中占有很重要的地位,在这一章中将讨论各种算法的分类,然后具体分析相应典型的算法。

### 5.1 信息隐藏算法概述

在设计具体的数据嵌入算法时,一般要考虑嵌入位置、嵌入数据量、嵌入方法、嵌入强度等要素,所以信息隐藏技术的算法一般分为替代算法、信号处理算法、编码算法、统计算法和伪装。在替代算法中一般包括位平面算法(bit plane methods)和基于调色板的算法(palette-based methods);信号处理算法中包括变换算法(transform methods)和扩频技术(spread spectrum techniques);编码算法中包括量化(quantizing)、抖动(dithering)和差错控制编码(error correcting codes);统计算法是使用假设与验证统计算法(hypothesis testing);伪装产生方法是用分形技术(fractals)。

在目前研究的信息隐藏算法中,主要集中于空域和变换算法。空域替代方法直接用秘密信息替代载体中的冗余部分。变换域可以分为DFT域、DCT域和Wavelet域,本章将详细讨论各种典型算法的原理,并给出相应的实验。

## 5.2 位平面算法

位平面算法是最早研究的一种算法,之所以研究位平面算法,是因为针对于压缩攻击以及统计分析等,必须结合JPEG等的核心算法,而JPEG 2000的核心编码算法是分数位平面算法。在JPEG 2000中,由Davod S. Taubman提出的具有最优化截断点的嵌入式块编码(embedded block coding with optimized truncation,EBCOT)算法实现中使用了位平面编码(BPC),建立的算法利用在位平面内或位平面间的对称和冗余,以便维护统计最小化,并且使BAC产生的可能编码比特流最小化。每个位平面的EBCOT有三个过程,在这些过程中的每一个过程中位平面部分的编码与其他两个过程都没有相互重叠。这就是位平面编码也称为分数(fractional)位平面编码的原因。在分数位平面的编码阶段中,将每个编码块分解成一定数量的位平面。如果子带的精确度是P位,那么在子带中的每个编码块被分解成P个位平面。位平面编码(bit-plane coding,BPC)应用于每个编码块的每个位平面上来产生以上下文形式的中间数据和二进制判定值。

每个位平面的EBCOT有三个过程,三个过程执行的次序如下所述。

- (1) 重要性传播过程(SPP)。在这个过程中,对位置的幅度值为1的位在第一时间进行编码,即相应采样系数的最重要位。
- (2) 幅度精炼过程(MRP)。在SPP中的位未被编码,而在前一个位平面(即当前位不是相应采样系数的最重要位)的幅度值是1,在这个过程中进行编码。
- (3) 清理过程(CUP)。位在上述两个过程中都没有被编码,那么将在这一过程中进行编码。这个过程也结合游程编码的形式,这将有助于零串的编码。

### 5.2.1 位平面算法概述

本小节中并不详细讨论位平面算法的具体编码方式,而是讨论这种算法在信息隐藏中的原理及实现。

因为位平面算法属于替代算法中的一类,所以位平面信息隐藏算法的处理方式一般分为如下几种:

- (1) 用秘密信息位代替图像的最低有效位。
- (2) 用秘密信息位代替图像的最低3位或4位。
- (3) 将秘密信息隐藏在图像的噪声中。

位平面方法能隐藏大量的秘密信息,但是这种方法对于一些针对图像处理的攻击而言,是很脆弱的。图5.1是位平面的示意图。

### 5.2.2 位平面算法实现

基于位平面示意图可知,G图像的每个像素占用8bit,将每个像素的特定位抽取出来,形成8个二值位平面图像,这种位平面分解可表示为:

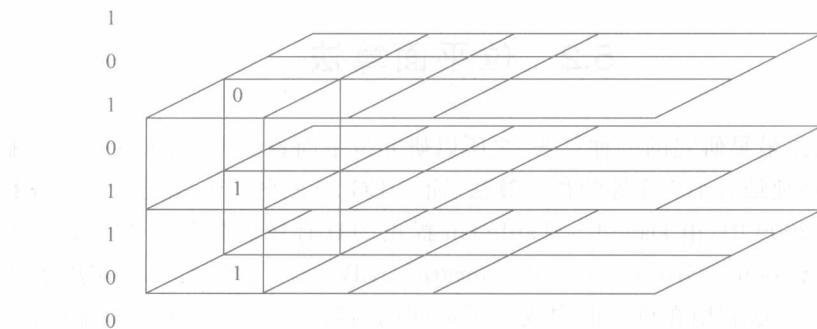


图 5.1 位平面示意图

$$G(x, y) = B_7(x, y) \times 2^7 + B_6(x, y) \times 2^6 + B_5(x, y) \times 2^5 + B_4(x, y) \times 2^4 \\ + B_3(x, y) \times 2^3 + B_2(x, y) \times 2^2 + B_1(x, y) \times 2^1 + B_0(x, y)$$

自然图像的高位位平面很重要, 图像的感知均在高位, 而低位位平面主要是图像的噪声。所以像前面所述, 一般可以替代最低有效位或者最多不超过替代最低的 4 位以内。

例: 以图像 lena.bmp 为例, 首先对图像进行位平面分解。

其源 Matlab 编码如下:

```
function g(action,varargin)
clear;
f = 'lena.bmp'; dim = 128;
n = fopen(f,'r');
img = fread(n,[dim,dim]);
mask = 8; % mask 的值范围是 1~8
a = bitshift(img, -1);
b = bitxor(img,a);
nim = bitget(b,mask);
imagesc(nim),colormap(gray)
```

实验结果如图 5.2 所示。



图 5.2 最高位平面到最低位平面

由实验结果可知,最高位平面代表了图像,而最低位平面一般是图像的冗余部分。一幅图像由表示像素的亮度值的矩阵组成,对于 $256 \times 256$ 的灰度图像,可以将图像分解成8个二值图像来表示8个位平面。

在图5.2的位平面分解图中,原始的自然图像都有相似的特征:高位位平面的轮廓特征强于低位位平面;并且随机性从高位到低位逐渐增强;另外最低位包含最少的图像信息,像素间的相关性是随机的,因而可用于图像信息的隐藏。

在位平面中最简单的嵌入是以确定的顺序直接嵌入到载体的数据位最低位里。但这样,就使得嵌入的数据位比较有规律地遵循着嵌入信息的分布规律,从而明显地改变了像素的统计规律,为检测提供较多统计信息。

为克服这一缺点可先以伪随机噪声序列将信息在嵌入前或嵌入的过程中进行处理。这样,信息就以伪随机噪声的形式存在于图像中,大大提高了检测的难度。这种方法嵌入简单,隐藏容量大,具有很好的不可感知性,然而对于鲁棒性以及抗干扰能力和其他一些安全问题比较大;对于任何形式的滤波以及处理相当敏感,比例的变化、旋转、剪切、噪声以及有损压缩都能够损坏隐藏图像;并且,攻击者可以通过简单地移除隐藏信息的位平面来彻底破坏信息。

那么最高的方式就是可以用随机性使低位的分布近似为高斯分布,为什么需要使用高斯分布呢,Cox等人提出高斯分布的随机序列具有更高的稳健性,可以更好地针对各类隐写分析和进攻。

随着位数的增高,分布趋向于平坦,零点值逐渐增大,位平面内的相关性逐渐增强。直方图可以很好地反映同一位平面间的像素的相关性,若像素的分布是随机的,直方图的分布会比较好地符合正态分布。位平面间的纹理的相似性表明在位平面中间存在着或多或少的相关性,众所周知,图像的最低位平面的分布是近似随机的,其与高位的相关性比较小,因而,将秘密信息预处理为随机序列。假设最低有效位随机信息的隐藏长度为L,因为随着位平面的降低,位平面的冗余度逐渐增加,故用于隐藏的位数呈逐渐递减趋势。

### 5.2.3 嵌入算法步骤和程序

位平面信息隐藏算法的具体操作步骤如下:

- (1) 首先用上述例子中的代码将图像分成8个位平面。
- (2) 将秘密信息预处理为伪随机序列,将秘密信息随机嵌入到最低和次低位平面中。

在这一步中使用第3章中的预处理来生成相应伪随机序列,然后与图像中的每个字节的高7位异或,将结果写入最低位。算法嵌入过程如图5.3所示。

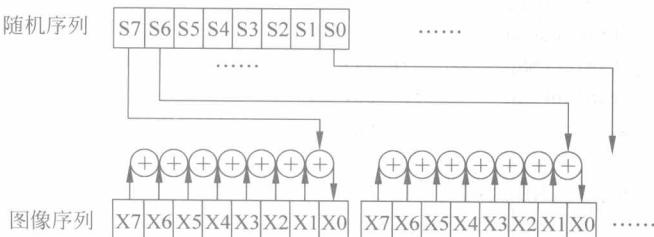


图5.3 算法嵌入过程

```

嵌入算法实现的程序代码：以下为用C语言实现的嵌入算法，该程序将信息隐藏于位图文件中。该程序首先读取位图文件头，然后根据文件头信息计算出隐藏信息的最大长度，接着从用户输入中读取信息，将其转换为二进制数据，并将其嵌入到位图文件中。最后，程序将修改后的位图文件写回磁盘。需要注意的是，该程序假设输入的信息长度不超过文件头中指定的最大长度，且输入的信息必须是偶数位长。

```

```

typedef struct tagBitMapFileHeader
{
    byte bfType[2];
    byte bfSize[4];
    byte bfReserved1[2];
    byte bfReserved2[2];
    byte bfOffBits[4];
}
int getValue(byte * A, int num)
{
    int result = 0;
    for(int i = num - 1; i>0; i--)
    {
        result += A[i];
        result = result<<8;
    }
    result += A[0];
    return result;
}
void hideInfo()
{
    FILE * fp;
    char data[MAX];
    char path[MAX];
    BitOperate operate;
    string path1 = "请输入图片路径(如: d:\\mypicture\\parrots.bmp)";
    cout<<path1<<endl;
    char temp3[100];
    cin.getline(temp3,100,'\\n');
    cin.getline(path,sizeof(path),'\\n');
    if((fp = fopen(path, "r +")) == NULL)
    {
        cout<<"打开文件的时候出现错误!"<<endl;
        return;
    }
    BitMap map = //存放文件头信息;
    (BitMap)malloc(sizeof(struct tagBitMapFileHeader));
    fread(map,sizeof(struct tagBitMapFileHeader ),1,fp);
    printf("请输入您要隐藏的信息:\n");
    cin.getline(data,MAX,'\\n');
    int dataLength = strlen(data);
    int mapLength = getValue(map->bfSize,4);
    int dataBegin = getValue(map->bfOffBits,4);
    if((dataLength * 8)>(mapLength-dataBegin))
    {
        printf("隐藏信息超过图片大小\n");
        return;
    }
    byte temp[32];

```

```
fseek(fp,databegin,SEEK_SET);
fread(temp,sizeof(char) * 32,1,fp);
int copy = datalength;
for(int i = 31; i>= 0; i--)
{
    bool v = (bool)(copy % 2);
    temp[i] = operate.bitSet(temp[i],1,v);
    copy = copy/2;
}
fseek(fp,databegin,SEEK_SET);
fwrite(temp,sizeof(byte) * 32,1,fp);
int times = 0;
byte ch[8];
while(times<DATALENGTH)
{
    fseek(fp,databegin + 32 + times * 8,SEEK_SET);
    fread(ch,sizeof(byte),8,fp);
    for(int i = 7; i>= 0; i--)
        ch[i] = operate.bitSet(ch[i],1,operate.bitAt(data[times],8 - i));
    fseek(fp,databegin + 32 + times * 8,SEEK_SET);
    fwrite(ch,sizeof(byte),8,fp);
    times++;
}
fclose(fp);
cout<<"信息隐藏成功!!"<<endl;
}
```

#### 5.2.4 实验和实验结果分析

下面是我们进行的实验结果的原始图像与嵌入秘密信息的图像,如图 5.4 所示,嵌入的秘密信息如图 5.5 所示。



图 5.4 原始图像与嵌入秘密信息的图像



图 5.5 嵌入的秘密信息

另外,在对信息隐藏算法进行度量时,经常用峰值信噪比和均方差,下面给出这两个参数的定义以及 Matlab 实现。

(1) 均方差(mean square error,MSE)。定义如下: 大小为  $M \times N$  的 256 色灰度图像  $f(i,j)$  和参考图像  $f_0(i,j)$ , 则图像  $f$  相对于  $f_0$  的 MSE 为:

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N (f(i,j) - f_0(i,j))^2}{MN}$$

Matlab 源程序如下:

```
function mse = MSE(Img1, Img0)
% 计算图像 Img1 对 Img0 的 MSE 值
% 输入参数
% Img1: 给定图像 Img1
% Img0: 参考图像 Img0
% 图像大小必须一样
% 输出参数 mse: 图像 MSE 值
N3 = size(Img1);
tmp = minus(uint8(Img1), uint8(Img0));
tmp = tmp. * tmp;
mse = sum(sum(tmp))/(N3(1) * N3(2))
```

(2) 均方根误差(RMSE)。定义为 MSE 的平方根,从而峰值信噪比(peak signal to noise ratio,PSNR)定义如下: 大小为  $M \times N$  的 256 色灰度图像  $f(i,j)$  和参考图像  $f_0(i,j)$ , 则图像  $f$  相对于  $f_0$  的 PSNR 为:

$$PSNR(f, f_0) = 10 \lg \frac{\max^2(f)}{MN \sum_{i=1}^M \sum_{j=1}^N (f(i,j) - f_0(i,j))^2}$$

在 256 色灰度图像计算中  $\max(f)$  通常用 255 代替,那么

$$PSNR(f, f_0) = 10 \lg \frac{255^2}{MN \sum_{i=1}^M \sum_{j=1}^N (f(i,j) - f_0(i,j))^2}$$

Matlab 源程序主要代码如下:

```
function psnr = PSNR(Img1, Img0)
% 计算图像 Img1 对 Img0 的 PSNR 值
% 输入参数
% Img1: 给定图像 Img1
% Img0: 参考图像 Img0
% 图像大小必须一样
% 输出参数
% psnr: PSNR 值
N3 = size(Img1);
tmp = minus(uint8(Img1), uint8(Img0));
tmp = tmp. * tmp;
MSE = sum(sum(tmp))/(N3(1) * N3(2));
if (MSE ~ = 0)
    psnr = 10 * log10(65025/MSE);
else
    psnr = 0;
```

```

psnr = 10000; % 标记为无穷大
end

```

图像之间越相似, RMSE 越小, PSNR 就越大。

使用该方法比较直观,秘密信息的嵌入符合从高位到低位的变换趋势,并且符合高斯分布,从实验结果可知,很难检测出秘密信息的存在。对原图像和嵌入秘密信息的图像的直方图分析如图 5.6 所示。

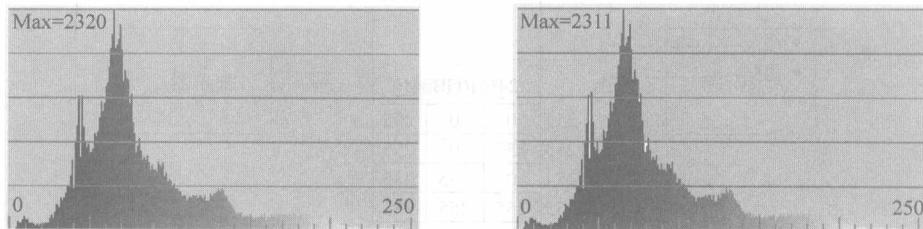


图 5.6 原图直方图与嵌入秘密图像后的直方图

#### 实验结果及分析:

在实验中,采用  $512 \times 512$  的 parrots 图像为载体图像。嵌入图像为 256 灰度级的 arnold 人脸图像。为增强可靠性,首先要对 arnold 图像进行预处理,可采用如下方法:

- (1) 直接嵌入和加密置乱后嵌入。
- (2) 采用伪随机方式在不同的位置嵌入秘密信息。在最低位平面随机嵌入数据,在最低的位平面和第二个最低位平面嵌入数据。
- (3) 通过直方图可知,图像变化很小,不易察觉和检测到嵌入的秘密信息,实现了信息隐藏。

## 5.3 调色板算法

在实际应用中,由于调色板图像占用空间少,广泛应用于网页、广告、彩信和图像处理等网络应用中,所以研究调色板算法具有现实和理论意义。调色板图像是指图像文件中包含调色板信息和图像索引信息,并能调用调色板中的颜色来显示图像。在索引图像中实现隐写有一定的难度,因为图像本身所使用的颜色数目有限,调色板的排列顺序不影响图像,但影响索引。索引图像包括一个数据矩阵  $X$ ,一个颜色映像矩阵 Map。其中 Map 是一个包含 3 列和若干行的数据阵列。Map 矩阵的每一行分别表示红色、绿色和蓝色的颜色值。索引色从像素值直接映射成颜色映射表,像素颜色由数据矩阵  $X$  作为索引指向矩阵 Map 进行索引。颜色映射表通常与索引图像存储在一起。

### 5.3.1 调色板算法原理

首先分析调色板的原理。不论是 GIF 格式和调色板 BMP 图像,都包含全局调色板数据块,它是以 RGB 方式描述全局调色板。还包含局部调色板数据,它也是以 RGB 方式描述

应用于局部某一帧的局部调色板。

调色板结构数组包含颜色三分量 RGB 和标志字段, 调色板的工作原理图如图 5.7 所示。

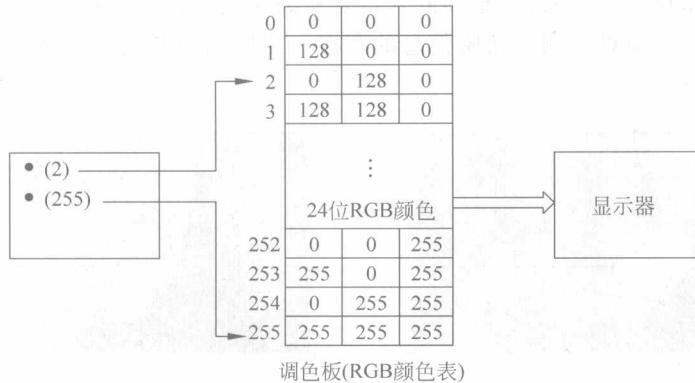


图 5.7 调色板工作原理

在调色板形成的 RGB 空间, RGB 色彩空间是一个不均匀的空间。均匀的色彩空间中两个颜色点间的欧几里得(Euclidean)颜色距离对应于人类视觉系统存在感知差异。另一方面, 在不均匀的色彩空间中, 在一部分色彩空间中, 距离为  $d$  的两种不同颜色表现出的感知差异在不同位置是不同的, 在成像应用中, 视觉均匀的色彩空间是很重要的。根据人类视觉生理知识可知, 不均匀的 RGB 空间需要映射为新的视觉均匀的空间。计算公式如下:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$$

在这个空间中, 各样本之间不是简单的线性关系, 假设样本为  $x_1, x_2, \dots, x_{256}$ , 那么  $x_i$  和  $x_j$  间的相似度测量应为:

$M_{ij} = \frac{D_{ij}}{D}$ , 这里  $D$  为最大的欧几里得颜色距离,  $D_{ij}$  是  $x_i$  和  $x_j$  之间的欧几里得颜色距离。为了保证图像的保真度,  $M_{ij}$  应该满足:  $M_{ij} < \delta, \delta \in [0, 1]$ 。

### 5.3.2 调色板信息隐藏算法实现

基于调色板处理算法就意味着改变代表图像颜色的彩色或灰度调色板; 基于调色板的算法中都利用了调色板的数据, 都使调色板数据发生了变化, 通过分析调色板原理可知, 对调色板数据的任何操作都会影响隐藏的秘密信息。

调色板算法使用时都会产生像素移位现象, 这种现象可能引起对隐藏数据的怀疑。为了避免上述问题, 本小节针对最低有效位的调色板方法进行了改进。对于最低有效位法的调色板方法, 许多工具都在使用 LSB 算法, 但是这种算法鲁棒性很差, 只需用简单的覆盖方法就能改变秘密信息。并且, 基于调色板的方法具有模板时, 可以被分析。在本实验中, 使用调色板类型图像, 在这种格式图像中隐藏秘密信息, 可以将秘密信息嵌入到调色板本身的色彩中, 也可以利用调色板中排列的次序来表示, 这两种方法嵌入秘密信息的容量受到了限制, 所以目前针对这种算法, 均是将秘密信息直接嵌入到每个像素的颜色值中。先减少调色

板中颜色的个数,然后产生新的颜色,产生的颜色总数小于调色板颜色数目的上限 256 色。根据嵌入秘密信息的容量,来随机地减少调色板中颜色的个数。

具体思路如下:

为了不引起怀疑,可以使用灰度级来克服颜色改变的问题,假定为彩色或 8 位图像,256 种不同颜色的顺序号为 0~255。为了嵌入信息,可以使用 S-Tools 将颜色从 256 减少到 32,并使用最低有效位 LSB 来隐藏秘密信息。在本算法中,在嵌入秘密信息前后,在视觉上没有太大改变,但是实际上位表示已经发生了变化。假定颜色  $c_i$  表示为 10110010,颜色  $c_i+1$  表示为 10110011,当根据像素颜色对的亮度进行分组时,人眼能分辨的颜色为  $L = 0.299R + 0.587G + 0.114B$ 。如图 5.8 和图 5.9 所示,parrots 是原始图像,hidden 是隐藏了秘密信息的图像。

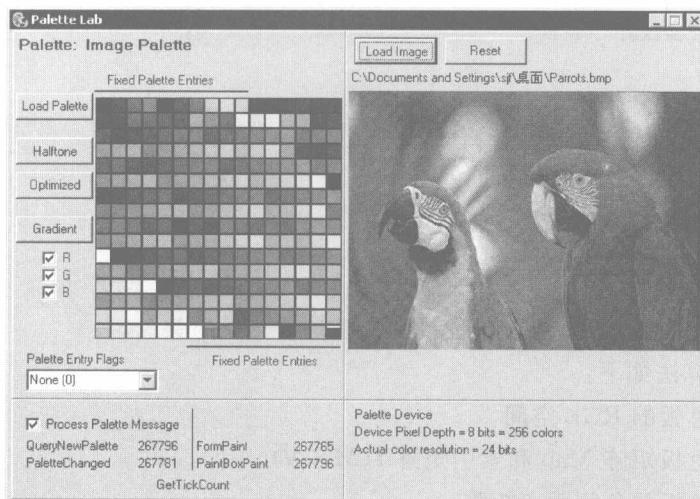


图 5.8 未隐藏秘密信息的调色板

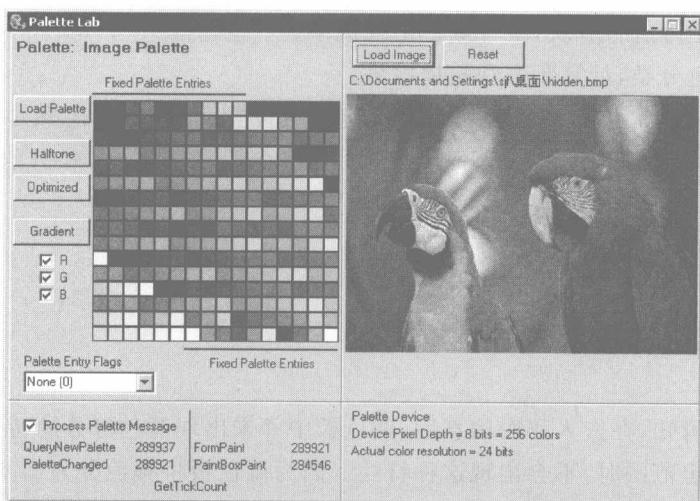


图 5.9 隐藏了信息的调色板

- 下面我们描述具体匹配颜色查找算法的实现过程：
- (1) 使用调色板的图像颜色量化实现过程。
  - (2) 遍历整个调色板，查找最接近的颜色。
  - (3) 计算源颜色与调色板颜色之间的差。
  - (4) 如果各种颜色比目前找到的所有颜色都接近源颜色，则使用该颜色。
  - (5) 如果是完全匹配的颜色，则跳出循环。

实现描述：

```

Begin
    定义源数据指针
    Repeat
        Int 32 * pSourcePixel
        遍历每一行
        查找最接近颜色
        For (int row = 0; row<height; row++)
        然后遍历每一列
        查找最接近颜色
    Until 完全匹配的颜色

```

这样通过这个算法程序，可以保证找到的总是最相近的颜色。

具体的嵌入算法如下：

- (1) 建立调色板的 RGB 空间。
- (2) 读入调色板矩阵 Map 和索引矩阵  $I(M \times N)$ 。
- (3) 计算调色板颜色间距离矩阵。
- (4) 对颜色分类。
- (5) 建立调色板分类索引。
- (6) 对图像矩阵进行分块，统计颜色出现概率。
- (7) 根据颜色分类，计算频率和扫描序号。
- (8) 由用户密钥来控制随机数发生器确定隐藏位置，设确定的隐藏位的调色板索引，判别奇偶，嵌入秘密信息，为 0 则不修改索引值，如为 1 则修改索引值表示已嵌入信息位。
- (9) 重复上述步骤直至隐藏了所有的信息。

### 5.3.3 调色板信息隐藏算法容量实验

由仿真实验可知，在嵌入秘密信息后，调色板中并未出现颜色群分的特性，并且，当嵌入的容量不超过一定范围时，不会出现这种特性。因为调色板将颜色分为 256 群，再由每群中选择最具代表性的一种颜色，所以共有 256 色，利用 3 位矢量来记录每个颜色的 RGB 值，将其存储，就形成了调色板，分析基于调色板的算法最佳容量，做了下面一组仿真实验。

嵌入秘密信息量占载体容量的 10%、30%、50%、70% 的调色板分别如图 5.10~图 5.13 所示。

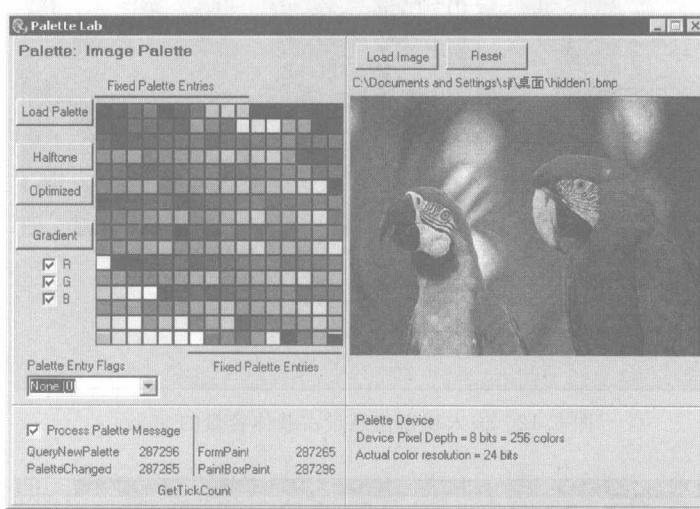


图 5.10 嵌入秘密信息量占载体容量的 10%

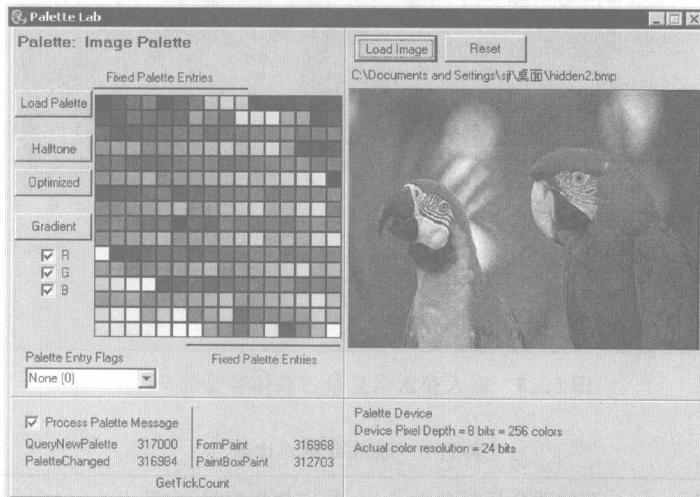


图 5.11 嵌入秘密信息量占载体容量的 30%

从上面的实验可知,当嵌入的容量很小时,调色板几乎不发生改变,也就是说,达到了隐写的目标,但当嵌入量超过 40% 之后,调色板发生了可以察觉的改变,分析这组实验可得出如下结论:

- (1) 嵌入秘密信息量少的时候,能起到很好的隐写效果。
- (2) 针对调色板来分析,只要将嵌入容量控制在 20% 以下,就能达到隐写的目标。

下面是上述实验选择不同容量载体时所进行的仿真实验,实验结果如表 5.1 所示。

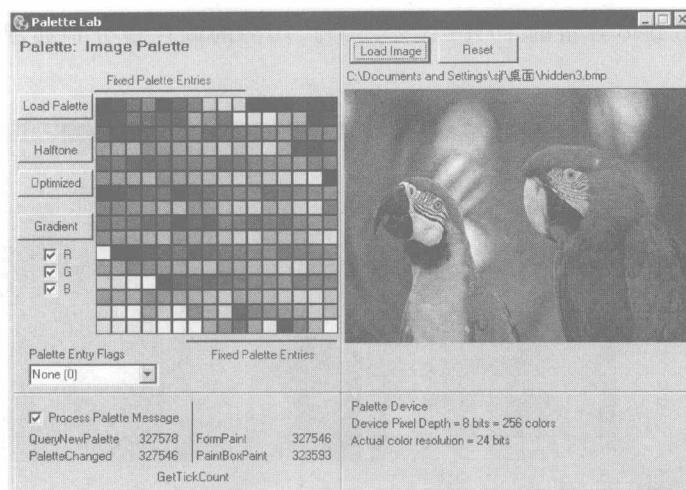


图 5.12 嵌入秘密信息量占载体容量的 50%

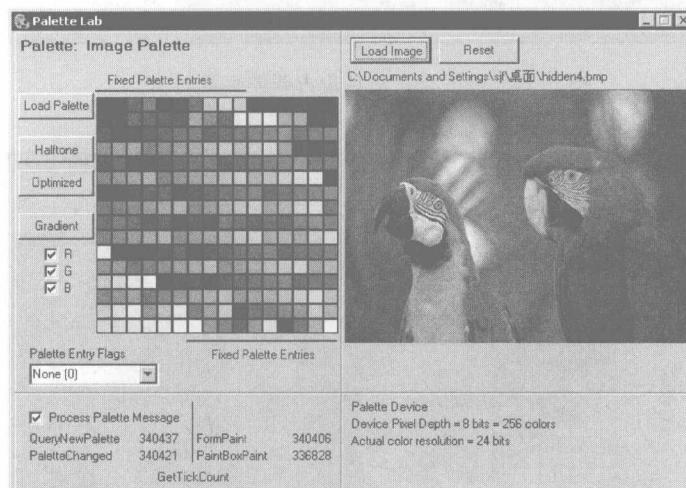


图 5.13 嵌入秘密信息量占载体容量的 70%

表 5.1 嵌入容量对比表

载体容量/字节	载体调色板颜色数	秘密信息容量/字节	所占比例
787496	254	294896	206248(70%)
772136	256	289136	195696(68%)
59864	256	22034	20834(95%)
503464	249	187484	159852(85%)
91064	256	33509	15556(46%)
787496	256	294896	211934(72%)
787496	254	294896	152076(52%)
787496	255	294896	6341(2%)
517864	256	193784	51729(27%)
189024	256	70053	1936(3%)
202656	256	75247	4768(6%)

表 5.1 的分析验证了上述的实验结论,当嵌入控制在 20% 的容量以内时,载体的调色板发生的变化几乎检测不到,这样就能达到信息隐藏的目标了。

另外,EzStego 是 Romana Machado 开发的基于调色板的隐写程序,它的基本原理是先对调色板进行分类,这样方便找到相近颜色进行替换,使颜色变化减到最小。读者可以下载 EzStego 进行测试。

## 5.4 空域信息隐藏算法

空间域算法使用像素的二维阵列来存放隐藏的数据,最典型的信息隐藏算法是最低有效位算法(least significant bit,LSB),它利用人类视觉的特点,对在某一阈值之下的变化感知不明显,这种技术使用嵌入秘密信息来替换原图像中的最低 N 位,如果只替换 1 位时,人眼对微小的颜色变化并不能感知,那么在图像处理时可以改变相邻像素间的差分来隐藏信息,这种算法广泛使用于流行的隐写软件中,许多隐写工具都使用最低有效位法,因为这种算法容量大并且容易实现,适用于大量应用。

信息隐藏最简单的域是空域,它是通过改变像素值来嵌入信息,并且改变后的像素值不影响原图像的统计信息。一般而言,嵌入信息对每个像素的影响至少要等于量化的值,这样这种算法才能不影响原图像的统计信息。

### 5.4.1 最低有效位算法原理

在我们要做的实验中,将用嵌入的信息替换最低位,在做实验之前,用简单的示例来说明其原理。

举例来说,11111111 是 8 位二进制代码,最右边的一位称为最低有效位(LSB),因为这一位的改变对这个数据的值影响最小。最理想的是每个字节的最低有效位都用其他信息代替时对整个文件的改变最小。这样秘密信息被分割并插入到图像文件的每个像素的最低有效位中。使用红绿蓝(RGB)模式的隐写工具生成图像调色板副本,也就是说生成 8 位图像。重新排列副本以便在 RGB 模式中相邻的颜色在调色板中也相邻。每个像素的 8 位二进制数值的最低有效位用秘密信息所代替。在调色板中,新的 RGB 颜色产生。隐写工具找到每个像素的 RGB 颜色的 8 位二进制代码。每个像素的 8 位二进制代码中的最低有效位是秘密数据文件的一位,然后,每一个最低有效位都写入到输出文件中。

8 位图像的例子:

1 像素

(00 01 10 11)

白 红 绿 蓝

嵌入 0011:

(00 00 11 11)

白 白 蓝 蓝

从这个例子可以看出,对于 8 位的图像,载体图像必须精心选择,因为 LSB 操作并不是任意的,因为颜色是有限的。对于 24 位图像,使用 LSB 方法在每个像素中可以隐藏 3 位秘密信息。

简单的 24 位图像的例子如表 5.2 所示。

表 5.2 24 位像素的例子

像素	红	绿	蓝
二进制数	10100001	10111010	11100011

对每个 8 位,二进制数的范围都是从 00000000~11111111,所以每个像素可以有  $256 \times 256 \times 256 = 16777216$  种可能的 RGB 组合。

下面举两个例子,手动完成数据的嵌入:

1 个像素

(00100111 11101001 11001000)

嵌入 101:

(00100111 11101000 11001001)

红 绿 蓝

嵌入 9 位数据 101101101 的例子,下面是 RGB 模式的编码:

10010101 00001101 11001001

10010110 00001111 11001010

10011111 00010000 11001011

从左到右,从上到下,嵌入的 9 位数据代替了原图像中的 3 个像素。

LSB 嵌入的操作适用于灰度图像。它可以将数据嵌入到最低有效位或者最低的前一位中,这样人眼不能察觉到图像的变化。但是 LSB 嵌入算法极易受到如剪切和压缩的影响,例如,原始图是.bmp 或者.gif 的图像,当进行无损压缩转换成.jpeg 格式的图像,然后再转换回原始格式时,将损坏数据中的最低有效位,从而使秘密信息不能在接收方还原。

#### 5.4.2 最低有效位算法实验

从上面的示例可知,对于 8 位图像的载体图像可以替换最低 1~N 位,用嵌入的低分辨率像素位来替换原图像中高分辨率的像素位,在提取时能够提取隐藏的信息,实验的载体图像如图 5.14 所示。

在嵌入时我们需要通过 Matlab 定义自己所使用的嵌入和提取函数,以及相应的 M 文件。

```
function [hidden] = lsb(cont,steg,imbed) % 嵌入函数
cont = double(cont); % cont 是需要调此函数时的原图像
steg = double(steg); % steg 是需要调此函数时需要嵌入的图像
stegshift = bitshift(steg, -(8 - imbed)); % 右移图像来嵌入
contprep = cont;
for i = 1: imbed
    contprep = bitset(contprep,i,0);
end
function [extract] = lsbImExt(hidden,imbed) % 提取函数
ext = uint8(hidden);
```



图 5.14 DEER

```

ext = bitshift(ext,8-imbed);
extract = double(ext);

clear;
load LSBImages;

lenaHide = lsbImHide(praying,airstrip,3);
showpic(lenaHide);
title('LenaHide Image');

lenaExt = lsbImExt(lenaHide,3);
showpic(lenaExt);
title('Extracted lena Image');

```

LSB 方法除了可以用于图像嵌入图像之外,还可以将文本信息嵌入载体中。我们经常使用  $256 \times 256$  大小的图像作为载体。为了完成这项实验,还需要编写 text.m 文件,用于完成在载体中文本的嵌入,可以替代 1 位,2 位和 3 位,只是所编写的 M 文件有所不同。嵌入的文本可以任意选择。

### 5.4.3 Hide and Seek 隐写软件分析与实验

在常用的隐写软件中,有许多使用 LSB 替代方法,如 Hide and Seek 等均使用最低有效位算法,平均而言,LSB 只需要改变图像一半左右的像素,将秘密信息嵌于最低位和次最低位,人眼是不能察觉的。

下面分析 Hide and Seek 中使用的 LSB 算法,此算法是基于私钥的。私钥的产生是通过伪随机置换产生,因此需要伪随机数置换发生器。根据 Kerchoff 的原则,发生器必须是安全的,并且任何人如果不知道密钥  $K$ ,就不可能猜测到伪随机序列。如果载体中  $N$  位可用,并且伪随机数是从  $0 \sim N-1$ 。如果欲隐藏的信息为  $n$  位,可以将秘密信息嵌入到载体中。因此,秘密信息随机嵌入到整个载体中。这种处理方式中的密钥  $K$  可以认为是一个黑盒子,对于不同的密钥值,它能产生不同的不可预测的序列,也就是说,具有密钥时,可以通过 hash 函数产生,参数为  $(i=0, \dots, n-1)$ ,函数的表示为:  $f_K(i) = H(K \circ i)$ 。因此得到了伪随机函数  $f_K(i), a \oplus b$  可以完成这个任务,因为一个代表密钥,一个代表参数,这两个部分由  $Y$  和  $X$  来表示,将  $K$  分为 4 个部分,  $K_1, K_2, K_3$  和  $K_4$ ,所以:

$$\begin{aligned}
Y &= Y \oplus f_{K_1}(X) \\
X &= X \oplus f_{K_2}(Y) \\
Y &= Y \oplus f_{K_3}(X) \\
X &= X \oplus f_{K_4}(Y) \\
&\text{return } Y \circ X
\end{aligned}$$

运行这个算法  $2^{2l-1}$  次,将会产生伪随机序列,这种方案与伪随机发生器产生的伪随机数一样安全。下面进一步介绍这种算法,图像  $I$  的维数为  $x$  和  $y$ ,第  $i$  个隐藏位的序列是:

$$Y = i \text{ div } x$$

$$X = i \bmod x$$

$$Y = (Y + f_{K_1}(X)) \bmod y$$

$$X = (X + f_{K_2}(Y)) \bmod x$$

$$Y = (Y + f_{K_3}(X)) \bmod y$$

return  $Y^* x + X$

返回值  $Y$  和  $X$  是相匹配的,这样就可以嵌入第  $i$  位的秘密信息位了,即  $K_1 \circ K_2 \circ K_3 = K$ 。如果这里有一幅  $800 \times 600$  的图像,将要嵌入的秘密信息为 1KB,密钥为:  $K=123456789$ ,那么载体图像  $N$  为 480000 位,而隐藏信息为  $1024 \times 8=8192$  位。这就是说只有少于 2% 的像素改动。对原始图像的改变量很小,并且没有视觉的改变。所以第 500 位的隐藏位置的算法如下:

$$Y = 500 \bmod 800 = 0$$

$$X = 500 \bmod 800 = 500$$

$$Y = (0 + H(123 \circ 500)) \bmod 600 = 7566 \bmod 600 = 366$$

$$X = (500 + H(456 \circ 366)) \bmod 800 = (500 + 3562) \bmod 800 = 62$$

$$Y = (366 + H(789 \circ 62)) \bmod 600 = (366 + 1563) \bmod 600 = 129$$

$129 \times 800 + 62$  的结果就是第 500 位隐藏的位置,即将秘密信息的第 500 位嵌入到  $X$  为 129,  $Y$  为 62 的像素的最低位或倒数第二位中。这种算法的缺点也是隐藏信息的容量很小,也就是说,嵌入的信息量与载体大小有直接关系,并且与相邻像素间的距离也有关系。为了使算法更安全,可以通过差错检测机制来增加额外一层。但这种算法的局限性在于只能用于无损压缩算法中。当将加密和隐写术结合起来时,将能达到很理想的安全状态。

众所周知,图像文件是很大的,为了存储和传输的需要,一定需要有损或无损压缩,BMP 和 GIF 图像使用无损压缩算法,压缩后图像与原图像相同,JPEG 使用有损压缩算法,压缩后图像与原图像并非完全相同。

提取隐藏位的一种方法可以通过 AND 操作来完成。1 代表隐写的位置,由于  $0 \text{ AND } 1 = 0$ ,并且  $1 \text{ AND } 1 = 1$ ,所以选择 1 作为操作。表 5.3 给出如何从 24 位图像中提取 Red、Green 和 Blue 的最低有效位。

表 5.3 提取最低有效位的 RGB 位(1 位 R,1 位 G,1 位 B)

像素	红	绿	蓝
二进制数	10100001	10111010	11100011
掩码	00000001	00000001	00000001
像素与掩膜	00000001	00000000	00000001

因为在 LSB 算法中,最多可替代每个像素中的 4 个元素,下面给出相应的表 5.4。

表 5.4 提取最低有效位的 RGB 位(2 位 R,4 位 G,3 位 B)

像素	红	绿	蓝
二进制数	10100001	10111010	11100011
掩码	00000011	00001111	00000111
像素与掩膜	00000001	00001010	00000011

24 位图像中的任何像素都可以使用,每个像素都具有 256 种 Red,256 种 Green 和 256 种 Blue 的值。通过嵌入,根据上表,两位 Red 可能产生的红色值为  $2^2=4$  种之一,对于 4 种 Green,则有 16 种可能,而对于 3 种 Blue,则有 8 种可能。那么  $2+4+3=9$  位,则可能有

512 种可能的 RGB 组合。

在 LSB 算法中,如果只替代最低位时,在进行对照时,有时信息量很大时,感觉低位变黑,这时,可以将嵌入位左移来使图像恢复原有亮度,LSB 举例如表 5.5 所示。

表 5.5 LSB 举例

像素	红	绿	蓝
二进制数	10100001	10111010	11100011
掩码	00000011	00001111	00000111
像素与掩码	00000001	00001010	00000011
掩码移位	11000000	11110000	11100000
提高亮度	01000000	10100000	01100000

这种算法的改进,理论上是可行的,但实际操作时,会给嵌入信息的提取造成困难,另外,移动的位数不宜过多,过多会引起图像质量的下降。下面基于这种考虑,又提出亮度/对比度的提高,这种方法需要记录每个 RGB 的最大和最小值。例如,亮度/对比度提高 Green 可以通过以下公式计算:

新 Green 值 =  $255 \times (\text{原有 Green 值} - \text{最小 Green 值}) / (\text{最大 Green 值} - \text{最小 Green 值})$ , 改变 Red, Green 和 Blue 将产生伪装的颜色,假设 2 位的 Red 范围是 0~3,4 位 Green 范围是 1~10,3 位 Blue 范围是 1~6。表 5.6 给出了相应的计算以及新生成的伪装色,伪装颜色对比度提高选项如表 5.7 所示。

表 5.6 新颜色的计算表

$$\begin{aligned}\text{NewRed} &= 255 \times (1-0)/(3-0) \\ \text{NewGreen} &= 255 \times (10-1)/(10-1) \\ \text{NewBlue} &= 255 \times (6-1)/(6-1)\end{aligned}$$

表 5.7 伪装颜色对比度提高选项

像 素	红	绿	蓝
像素与掩码(二进制)	00000001	00001010	00000011
像素与掩码(十进制)	1	10	3
提高对比度(十进制)	85	255	102
提高对比度(二进制)	01010101	11111111	01100110

下面是根据最低有效位算法嵌入文档的对比图像。



图 5.15 原始图像与嵌入秘密信息的图像

## 5.5 频域变换信息隐藏算法

在前面的研究中,我们知道基于调色板算法、位平面和空间域算法,虽然存在着容量大,容易实现及计算复杂度低等优点,但都存在着易受攻击的弱点,任何有意的攻击都会破坏隐写的信息,从而使隐写失败,达不到隐写的目的。目前大部分研究都集中在频域中,因为基于变换域的技术可以嵌入大量数据,并能保持很高的不易察觉性,这类变换一般都基于图像变换,并且可以基于局部或全局。下面先介绍一下变换域技术。

一般均用二元函数  $f(x, y)$  作为图像的数学表示。理解图像变换实质上是指把图像变成另一种数学表示方式的操作,通过变换改变图像的表示域及表示数据,将原定义在图像空间的图像以某种形式转换到另外一些空间,并利用这些空间特性来进行图像处理。变换域技术包括:离散余弦变换(discrete cosine transform, DCT)、离散小波变换(discrete wavelet transform, DWT)、离散傅里叶变换(discrete Fourier transform, DFT)和 Mellin 傅里叶变换(mellin-Fourier transform)。与此内容相关的有大量数学基础。在本小节介绍 DFT 和 DCT 技术的基本理论与相应算法,在 5.6 节中介绍 DWT 的基础理论和相应算法。

### 5.5.1 离散傅里叶变换 DFT

空间域中是通过对图像的数据的像素值进行运算来实现信息隐藏,而在频域中是对图



图 5.16 lena

像的全局的频率数据进行编码来实现信息隐藏,频域中的方法具有很强的健壮性。无论是何种变换都是将图像从空域变换到频域,使我们能够定量地分析其特点。本小节使用的载体图像如图 5.16 所示。

对于计算机而言,最常用的是二维离散傅里叶变换,如下所示:

$$f(u, v) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

逆变换公式如下所示:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

频谱公式如下所示:

$$\begin{aligned} F(u, v) &= |F(u, v)| e^{j\varphi(u, v)} = R(u, v) + jI(u, v) \\ |F(u, v)| &= [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}} \end{aligned}$$

下面进行具体的二维离散傅里叶变换,首先对矩形函数进行傅里叶变换,其具体 Matlab 代码如下:

```
Clear
N = 100
f = zeros(50,50);
figure(1)
```

```

imshow(f,'notruesize')
F = fft2(f,N,N);
F2 = fftshift(abs(F));
figure(2)
X = 1:N; y = 1:N;
mesh(x,y,F2(x,y));
colormap(gray); colorbar
Xm = abs(F);
imagesc(fftshift(Xm))
imagesc(fftshift(log(Xm)))

```

程序运行结果如图 5.17 和图 5.18 所示。

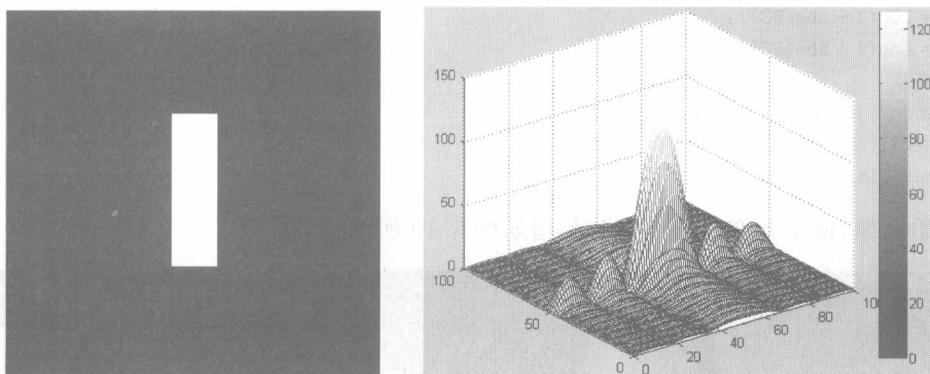


图 5.17 矩形连续函数及其傅里叶变换幅值

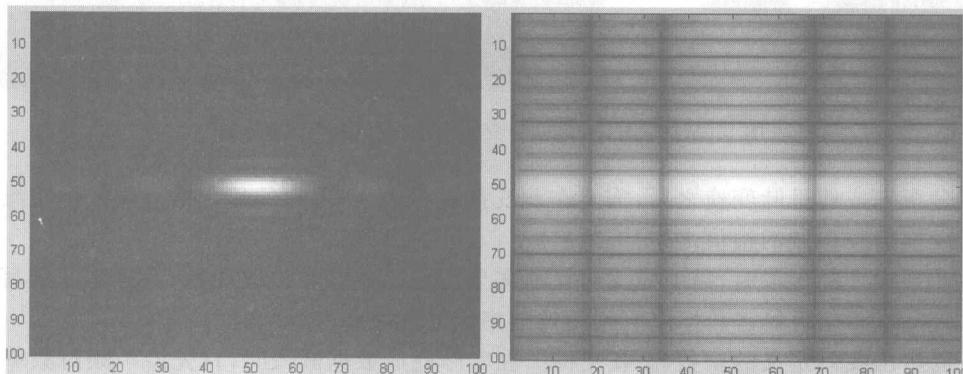


图 5.18 傅里叶变换频谱及傅里叶变换幅值对数图

下面对图像进行傅里叶变换，具体步骤和 Matlab 代码如下：

- (1) 读原始图像。
- (2) 利用 `fft2` 对原始图像进行快速傅里叶变换。
- (3) 利用 `abs` 函数得到傅里叶频谱。
- (4) 利用 `imshow` 来可视化频谱图像。
- (5) 利用函数 `fftshift` 将变换后的图像原点移动到频率矩形的中心。
- (6) 显示变换了中心后的频谱图。

- (7) 将结果利用对数变换进行处理。  
 (8) 利用傅里叶逆变换的实部值恢复原始的图像。  
 提取结果的实部输出。

```
f = imread('lena.bmp') * 注意,这里此图像已在 matlab 的 work 子目录中 *
X = fft2(x);
imshow(X);
Xm = abs(X);
Xa = angle(X);
FC = fftshift(Xm)
Figure, imshow(FC,[])
或者 imagesc(fftshift(log(FC)))
S2 = Log(1 + abs(FC));
S2 = Log(1 + abs(FC))
反变换:
xrecon = ifft(Xm. * exp(i * Xa));
g = real(ifft2(F))
figure, imshow(g);
```

原始图像、傅里叶变换的图像和频谱如图 5.19 所示。

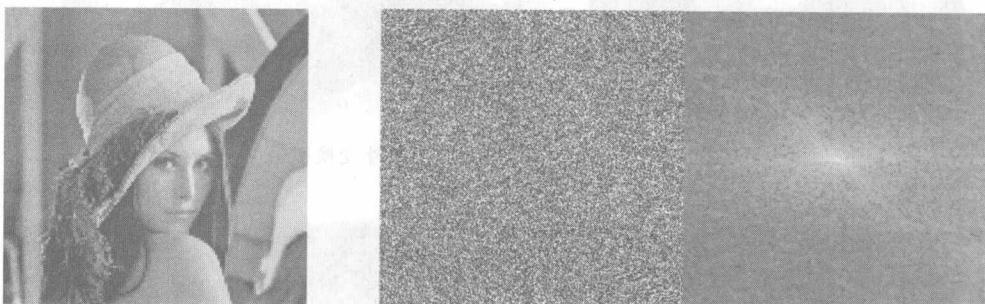


图 5.19 原图像、傅里叶变换图像和傅里叶变换频谱

在分析完成傅里叶变换之后,需要在此变换过程中,将用户的秘密信息嵌入到二维离散傅里叶变换后频率域的幅值之中。

在实现嵌入和提取时,需要定义自己所需要的嵌入函数和提取函数,并且需要编写相应的 M 文件,来实现将秘密数据嵌入到图像频率域的幅值中。

下面给出我们定义的函数和 M 文件。在实际完成数据嵌入时,首先沿中频对频率域数据进行编码,使用中频的原因在于它对图像质量影响最小。自己定义的.m 文件将通过输入的梯度值(Alpha)来生成尺度二进制数据向量,Alpha 的取值范围为 10000 ~ 25000。Alpha 的值既不能太大也不能太小,太大则会严重影响图像的保真度,如果太小算法的健壮性又会很差。在傅里叶变换中,此处所述算法使用了幅值,另外有兴趣的读者还可以使用相位。

- (1) 首先,计算图像(parrots)的幅值。范围是 $-\pi \sim \pi$ ,结果如图 5.20 所示。
- (2) 从变换后的图像可知,图像的主要内容由低频系数决定,高频部分形成图像的边缘和轮廓。为了使图像改变量最小,应该把秘密信息嵌入到中频中。根据人眼的最小视察差,最易观察到低频的改变,而如果要改变高频,则会出现局部边缘的突然变化。首先将秘密信

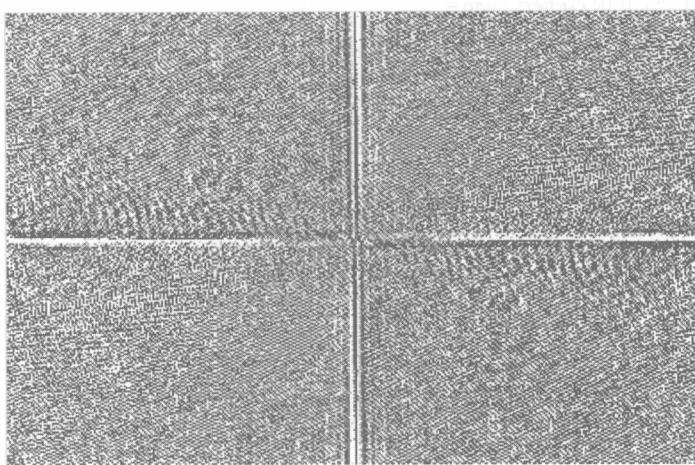


图 5.20 二维 fft 变换后的幅值

息转换成二进制流(可以通过预处理部分的 RC4 等算法或在 Matlab 中自己编写函数来完成),输出的是表示这些秘密信息的行向量,然后构造一矩阵,大小与 fft 相同,并初始化为 0。

这里给出一个示例。

```
function [text] = btotext(b)
len = length(b);
output_str = '';
char_temp = '';
for x = 0 : 6 : len - 6,
    sub_string = num2str(b(x + 1 : x + 6));
    switch sub_string
        case {'0 0 0 0 0 0'},char_temp = '';
        case {'0 0 0 0 0 1'},char_temp = 'a';
        case {'0 0 0 0 1 0'},char_temp = 'b';
        case {'0 0 0 0 1 1'},char_temp = 'c';
        case {'0 0 0 1 0 0'},char_temp = 'd';
        case {'0 0 0 1 0 1'},char_temp = 'e';
        case {'0 0 0 1 1 0'},char_temp = 'f';
        case {'0 0 0 1 1 1'},char_temp = 'g';
        case {'0 0 1 0 0 0'},char_temp = 'h';
        case {'0 0 1 0 0 1'},char_temp = 'i';
        case {'0 0 1 0 1 0'},char_temp = 'j';
        case {'0 0 1 0 1 1'},char_temp = 'k';
        case {'0 0 1 1 0 0'},char_temp = 'l';
        case {'0 0 1 1 0 1'},char_temp = 'm';
        case {'0 0 1 1 1 0'},char_temp = 'n';
        case {'0 0 1 1 1 1'},char_temp = 'o';
        case {'0 1 0 0 0 0'},char_temp = 'p';
        case {'0 1 0 0 0 1'},char_temp = 'q';
        case {'0 1 0 0 1 0'},char_temp = 'r';
        case {'0 1 0 0 1 1'},char_temp = 's';
```

```

case {'0 1 0 1 0 0'},char_temp = 't';
case {'0 1 0 1 0 1'},char_temp = 'u';
case {'0 1 0 1 1 0'},char_temp = 'v';
case {'0 1 0 1 1 1'},char_temp = 'w';
case {'0 1 1 0 0 0'},char_temp = 'x';
case {'0 1 1 0 0 1'},char_temp = 'y';
case {'0 1 1 0 1 0'},char_temp = 'z';
case {'0 1 1 0 1 1'},char_temp = '0';
case {'0 1 1 1 0 0'},char_temp = '1';
case {'0 1 1 1 0 1'},char_temp = '2';
case {'0 1 1 1 1 0'},char_temp = '3';
case {'0 1 1 1 1 1'},char_temp = '4';
case {'1 0 0 0 0 0'},char_temp = '5';
case {'1 0 0 0 0 1'},char_temp = '6';
case {'1 0 0 0 1 0'},char_temp = '7';
case {'1 0 0 0 1 1'},char_temp = '8';
case {'1 0 0 1 0 0'},char_temp = '9';
case {'1 0 0 1 0 1'},char_temp = '.';
case {'1 0 0 1 1 0'},char_temp = '?';
case {'1 0 0 1 1 1'},char_temp = '';
case {'1 0 1 0 0 0'},char_temp = ';';
case {'1 0 1 0 0 1'},char_temp = '-';
case {'1 0 1 0 1 0'},char_temp = ',';
otherwise char_temp = '!';
end
output_str =[output_str char_temp];
end
text = output_str;

```

(3) 编写自己的 M 文件,完成秘密信息的嵌入,在频域,我们选择了中频,沿中频这个区域,可以选择为一个圆,圆的半径由用户来选择。另外还需要尺度 alpha 的选择。

下面是一个函数示例,这个函数将数据信息嵌入到了图像频域的幅值中。

```

function [out_pic_mat] = freqmark(in_pic_mat,in_msg,r,alpha)
bin_msg = text2bin(in_msg);
bin_msg = [bin_msg zeros(1,r - length(bin_msg) - 4)];
watermark = alpha * bin_msg;
[N,M] = size(in_pic_mat);
circ = zeros(N,N);
fft_imagem = fftshift(fft2(in_pic_mat));
mag_imagem = abs(fft_imagem);
phase_imagem = angle(fft_imagem);
k = 1;
C = N/2 + 1;
for x = C + 1: C + r - 4
    A = x - C;
    B = round(sqrt(r^2 - (x - C)^2));
    circ(x,C + B) = watermark(k); % x = c + A
    circ(C - A,C - B) = watermark(k);
    circ(x,C - B) = watermark(k);
    circ(C - A,C + B) = watermark(k);
    k = k + 1;
end
out_pic_mat = ifft2(circ);

```

```

k = k + 1;
end
mag_imagem_marked = mag_imagem + circ;
for i = 1 : N,
    for j = 1 : M,
        if (circ(i,j) == 0)
            mag_imagem_marked(i,j) = mag_imagem_marked(i,j) - alpha/10;
        end
        imagem_marked_freq(i,j) = mag_imagem_marked(i,j) * (cos(phase_imagem(i,j)) + sqrt(-1)
* sin(phase_imagem(i,j)));
    end
end
imagem_marked_freq = fftshift(imagem_marked_freq);
out_pic_mat = real(ifft2(imagem_marked_freq));

```

(4) 嵌入结果如图 5.21 所示。



图 5.21 原图像与嵌入秘密信息后的图像

通过上述实验可知,使用频域来隐藏数据是非常有效的方法,但算法的复杂度增高。频域中的算法比在空域中的算法具有更好的鲁棒性,并且扩大了数据隐藏的容量。

### 5.5.2 离散余弦变换 DCT

DCT 的使用范围很广泛,可以使用 DCT 进行信息隐藏,将信息嵌入载体而非噪声,可以对处理后的伪装载体进行重建,来比较嵌入信息后的不同,可以用于压缩域,主要是对 JPEG 格式图像的压缩,DCT 具有能使图像的最重要信息集中在 DCT 的几个系数上的性能,载体或嵌入的图常被分成  $8 \times 8$  块或  $16 \times 16$  的小块,对每一小块分别计算其二维 DCT,DCT 系数值经过计算后被量化、编码和传输。在接收方,对量化的 DCT 系数进行解码,并将各小块重构成一幅图像。

下面首先介绍要用到的关于 DCT 和 JPEG 算法的基础知识。

JPEG(joint photographic experts group)是由 ISO 和 IEC 两个组织机构联合组成的一个专家组,负责制定静态的数字图像数据压缩编码标准,这个专家组开发的算法称为 JPEG 算法,既可用于灰度图像又可用于彩色图像。JPEG 专家组开发了两种基本的压缩算法,一

种是采用以离散余弦变换(discrete cosine transform, DCT)为基础的有损压缩算法,另一种是采用以预测技术为基础的无损压缩算法。

使用有损压缩算法时,在压缩比为 25 : 1 的情况下,压缩后还原得到的图像与原始图像相比较,非图像专家难以找出它们之间的区别,因此得到了广泛的应用。我们就是将信息隐藏在相应的 DCT 系数之上,达到信息隐藏的目的。

### 1. JPEG 算法

#### 1) 图像的差分编码方法

源图像为  $I(x, y)$ , 使用简单的差分计算可以定义差分图像  $D(x, y)$  为:

$$D(x, y) = I(x, y) - I(x - 1, y)$$

或者使用二维调和量算子计算来定义差分图像  $D(x, y)$  为:

$$D(x, y) = 4I(x, y) - I(x, y - 1) - I(x, y + 1) - I(x + 1, y) - I(x - 1, y)$$

因为在正常的图像  $I$  中存在着空间上的冗余,而差分图像具有很窄的直方图并且具有较小的熵。

构造差分的预测值,包括此前像素值的相邻的三个像素的值,如图 5.22 所示。进行编码: 编码器将当前的像素值与 X 位置的像素值进行比较,然后使用无损压缩技术对差值部分进行压缩。在这里可以使用 Huffman 编码方案,如图 5.23 所示。



图 5.22 嵌入秘密图像与差分图像

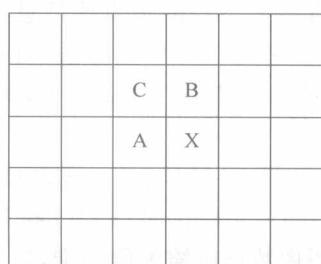


图 5.23 Huffman 编码方案

## 2) JPEG 算法的主要计算步骤

JPEG 压缩是有损压缩, 它利用了人的视觉系统的特性, 使用量化和无损压缩编码相结合来去掉视觉的冗余信息和数据本身的冗余信息。压缩编码大致分成三个步骤:

(1) 使用正向离散余弦变换(forward discrete cosine transform, FDCT)把空间域表示的图像变成频率域表示的图像。

(2) 使用加权函数对 DCT 系数进行量化, 这个加权函数对于人的视觉系统是最佳的。

(3) 使用赫夫曼可变字长编码器对量化系数进行编码。

注意译码(解压缩)的过程与压缩编码过程正好相反。

而在信息隐藏中就是需要将要隐藏的信息加入到 DCT 的系数中, 并且人的视觉系统感觉不到图像的变化。

JPEG 算法与彩色空间无关, 因此“RGB 到 YUV 变换”和“YUV 到 RGB 变换”不包含在 JPEG 算法中。JPEG 算法处理的彩色图像是单独的彩色分量图像, 因此它可以压缩来自不同彩色空间的数据, 如 RGB、YCbCr 和 CMYK。

图 5.24 给出了 JPEG 压缩编码-解压缩算法框图。

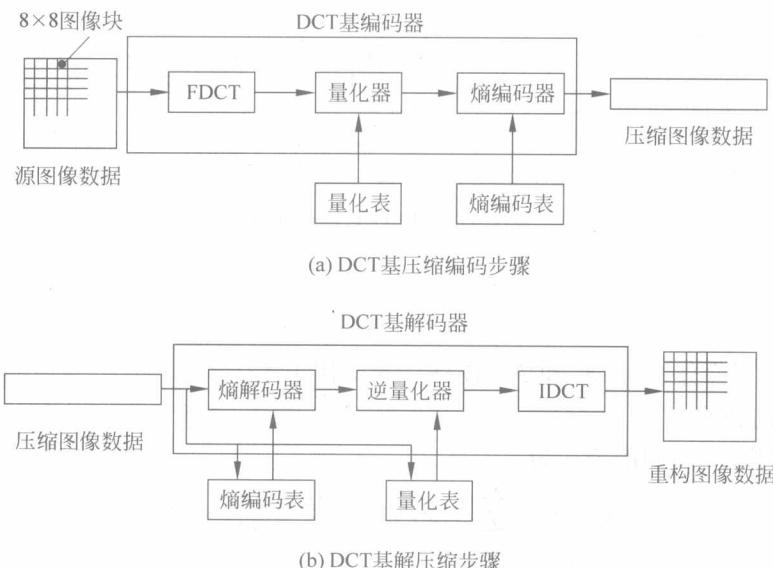


图 5.24 JPEG 压缩编码-解压缩算法框图

下面通过图示来介绍 JPEG 算法的实现, 如图 5.25 所示。

下面介绍图中用到的术语名词。

- FDCT 正向离散余弦变换。
- DPCM differential pulse code modulation, 使用差分脉冲编码调制对直流系数(DC)进行编码。
- RLE run-length encoding, 使用行程长度编码对交流系数(AC)进行编码。

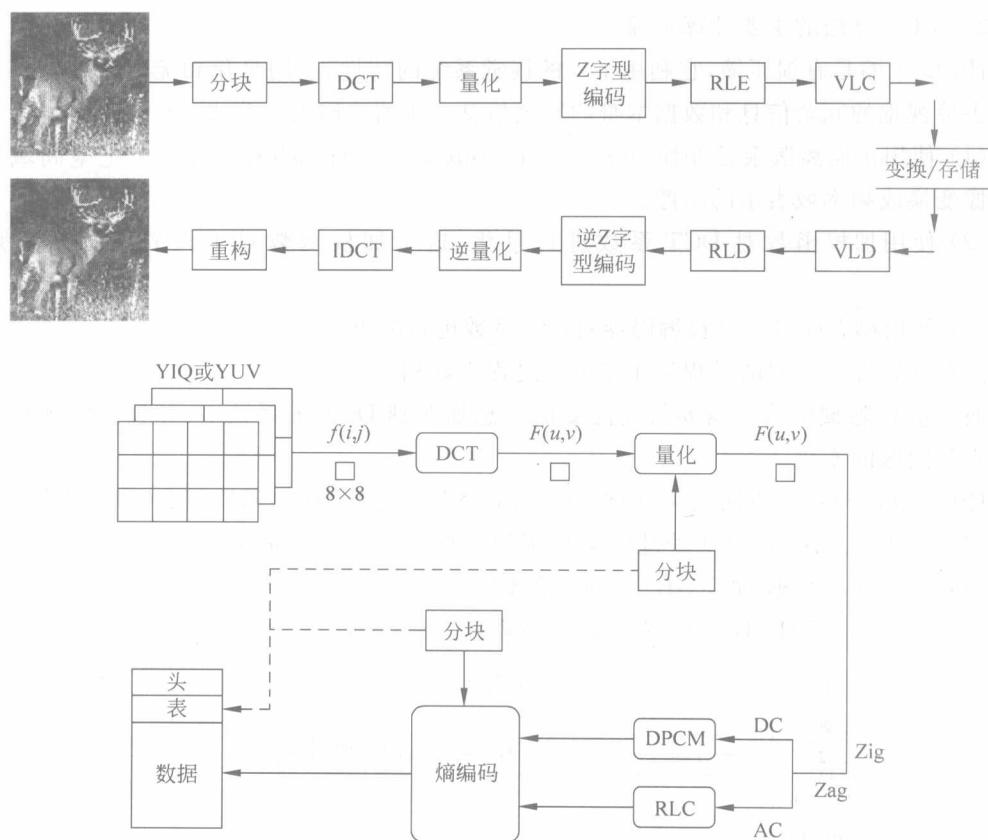


图 5.25 JPEG 算法的实现

## 2. DCT 基础

DCT 将源图像分解为 DC 和 AC 成分, IDCT 用于图像的重建。

### 1) DCT 的公式

对于原图像 A,用到的公式为:

$$b(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} a(x, y) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right)$$

变换为伪装图像 B 所用到的变换公式为:

$$a(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) b(u, v) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right)$$

在上述两个公式中:

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u \neq 0 \end{cases}$$

另外,DCT 分为一维 DCT 和二维 DCT,二维 DCT 公式和反变换公式如下:

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{i=0}^7 \sum_{j=0}^7 \cos\left(\frac{(2i+1)u\pi}{16}\right) \cos\left(\frac{(2j+1)v\pi}{16}\right) f(i, j)$$

$$\tilde{f}(i,j) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u)C(v)}{4} \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} F(u,v)$$

## 2) DCT 变换

下面对正向离散余弦变换(FDCT)作几点说明。

(1) 将图像分解成单独的彩色图像分量,如图 5.26 所示。



图 5.26 彩色图像分量分解

(2) 把整个分量图像分成  $8 \times 8$  的图像块,如图 5.27 所示,并作为二维离散余弦变换 DCT 的输入。通过 DCT 变换,把能量集中在少数几个系数上。 $f(i,j)$  经 DCT 变换之后, $F(0,0)$  是直流系数,其他为交流系数。

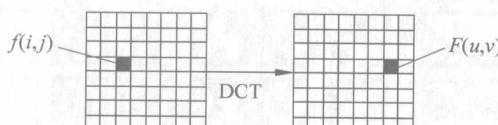


图 5.27 离散余弦变换

(3) 在计算二维 DCT 变换时,可使用下面的计算式把两维的 DCT 变换变成一维的 DCT 变换,如图 5.28 所示。

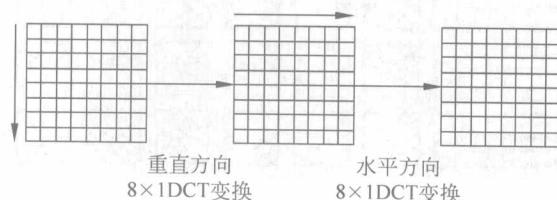


图 5.28 DCT 变换

(4) 将图像 A 转换成 YIQ 颜色空间, YIQ 中的 Y 代表亮度(luminance), I 代表色调(hue)Q 代表饱和度(saturation), 如图 5.29 所示。

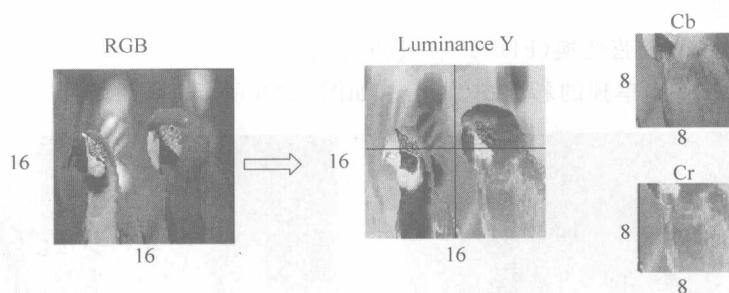


图 5.29 图像的 YIQ 空间转换

然后将每个颜色空间部分分成  $8 \times 8$  块, 对每一块使用 DCT 变换, DCT 分为一维和二维, 而图像往往是二维的。二维的 DCT 可以有两种方法: 一种是作为一种旋转; 另一种是作为一个  $N$  维矢量空间的基。第一种将  $n \times n$  像素块开始, 首先考虑把块中的每一行看成是  $n$  维空间的点  $(p_x, 0, p_x 1, \dots, p_x, n-1)$ , 利用下式的内层求和来旋转这些点, 得到一个  $n \times n$  的系数数据块  $G_{1,x,v}$ ,

$$G_{1,x,v} = C_V \sum_{y=0}^{N-1} a(x, y) \cos\left(\frac{\pi u(2x+1)}{2N}\right)$$

把上式的结果的列看成是  $n$  维空间的点, 然后旋转, 结果在块的左上角是一个大系数, 其他则是  $n^2 - 1$  的小系数。另外假设  $n=8$ , 可以生成如图 5.30 所示的 64 个  $8 \times 8$  的块, 作为 64 维矢量空间的基, 即基图像, 任何一个  $8 \times 8$  的像素块 B 都可以解释为基图像的线性组合, 而这个线性组合的权值就可以看成是 DCT 的系数。

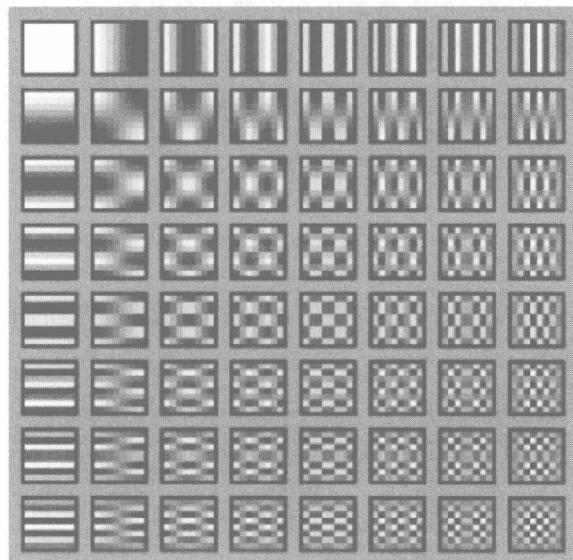


图 5.30 二维 DCT 的 64 个基图像

### 3. 块操作

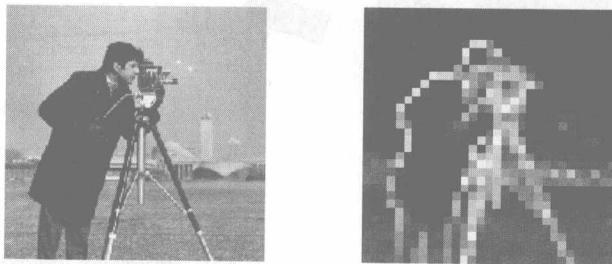
在 DCT 操作中,需要将图像进行分块,下面我们介绍一下块操作,块操作分为两类,滑块邻域操作和分块操作,前者通常用于滤波中,我们在 DCT 中使用的是分块操作,也就是将图像分为一个个相等的矩形块,一次单独处理一个块,超出图像区域以外的像素位置以 0 填充。

下面我们对图像 parrots 进行  $8 \times 8$  的分块。

下面是仿真实验的代码:

```
I = imread('cameraman.tif');
fun = inline('std2(x) * ones(size(x))');
I2 = blkproc(I,[8 8],fun);
subplot(121),imshow(I), xlabel('(a) 原图')
subplot(122),imshow(I2,[]), xlabel('(b) 块处理后的图像')
```

图 5.31 是仿真的结果。



(a) 原图

(b) 块处理后的图像

图 5.31 块处理前后图像比较

经过 DCT 变换后的  $8 \times 8$  系数矩阵的左上部分为直流部分,右下部分为高频部分。

```
I = imread('cameraman.tif');
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1 * x * P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   0   0   0   0   0   0
        1   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1. * x',mask);
I2 = blkproc(B2,[8 8],'P1 * x * P2',T',T);
imshow(I),figure,imshow(I2)
```

DCT 系数

0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

分析系数可知,图像主要集中在几个重要系数上,其他的系数接近 0 或者意义很小。下面用图形形式表示系数,如图 5.32 所示。可以更直观地看到,少数系数就能完成图像的重构。

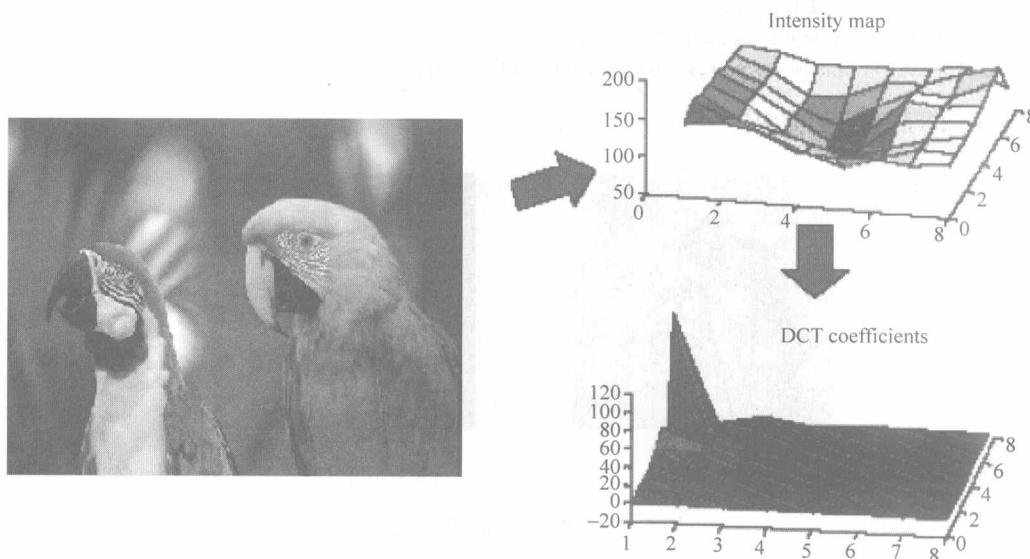


图 5.32 DCT 系数

由图像可知,图像中有用的图像内容改变相对缓慢,在小的区域如  $8 \times 8$  的图像块中,对很密集值的几次较大改变是很不常见的。在图像中的许多信息都是重复的,因此是空间冗余的,大多数块的重构只需要较小的几个重要系数,通常是最低的频率系数。

#### 4. 量化

在进行 DCT 分解之后,需要进行量化,首先介绍 JPEG 中的量化,然后再介绍信息隐藏中如何利用量化。

量化是对经过 DCT 变换后的频率系数进行量化。量化就是提取出有效的能代表图像的系数,并与预定的量化系数相除,所得的值由表列出,然后将所有的值近似等于整数,如图 5.33 所示。

量化的公式如下:

$$\hat{F}(u,v) = \text{round} \left( \frac{F(u,v)}{Q(u,v)} \right)$$

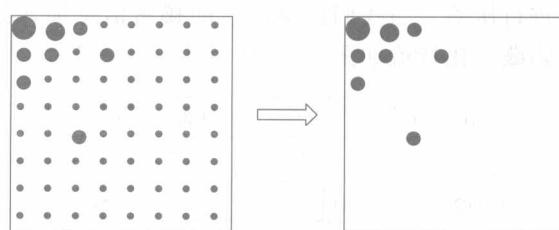


图 5.33 量化示意图

分子代表 DCT 系数,分母代表量化矩阵熵,所得结果为量化后的量化表值。也就是说,将 DCT 系数与预定的量化系数相除,然后将所有的值近似等于整数。

量化表如表 5.8 所示。

表 5.8 量化表

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

下面我们介绍，信息隐藏时如何在量化过程中，嵌入隐藏的秘密信息。当使用 DCT 进行信息隐藏时，发送方和接收方需要提前协商在每个  $8 \times 8$  块中的两个 DCT 系数的位置，假设协商时中频具有相同的量化值：如第(3,2)位置与第(4,1)位置的值，这样双方知道了 DCT 系数的位置。

接下来,将 DCT 应用到每个  $8 \times 8$  的块,用  $B_i$  表示,对每个块进行编码,用 1 位来表示,是 0 或者 1,如果要嵌入的信息位为 1,那么  $B_i(4,1)$  位置的系数与  $B_i(3,2)$  位置的系数进行比较,将比较大的系数放在  $(4,1)$  位置,如果信息位为 0,则将两位置中较小的系数放在  $(4,1)$  位置,如果  $|B_i(4,1) - B_i(3,2)| < \mu$ ,那么需要调整  $B_i(4,1)$  和  $B_i(3,2)$  的值,使  $|B_i(4,1) - B_i(3,2)| > \mu$ 。这个步骤可能会引起图像重构时图像的变化,所以目前许多方式已经对此进行改进。在提取数据时,在每块上执行 DCT,并且比较  $(4,1)$  位置和  $(3,2)$  位置的值,如果  $B_i(4,1) > B_i(3,2)$ ,那么信息位为 1,否则为 0。

量化之后进行 Z 字形编排,如图 5.34 所示,频率较低的系数放在矢量的顶部。然后将

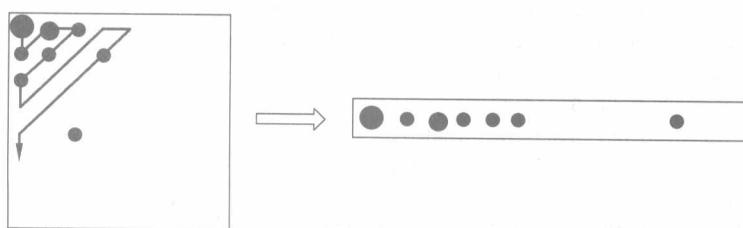


图 5.34 Z字形编排

所有的位组成位数据流进行压缩。当我们把要嵌入的秘密信息加入到量化后的系数之后，我们就已经完成了信息隐藏。比特流如图 5.35 所示。

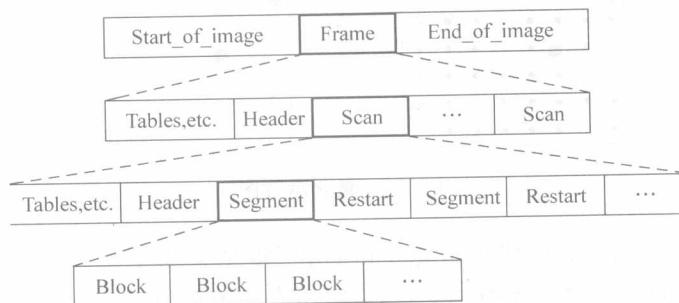


图 5.35 比特流

下面给出具体的实现 DCT 嵌入的代码：

```
dct_high.m
clear;
clc;
a0 = imread('D:\work\pic\lena.bmp');
a0 = rgb2gray(a0);
[r,c] = size(a0);
k = (r * c / 64);
da0 = blkproc(a0,[8,8],'dct2');
ca0 = im2col(a0,[8,8],'distinct');
cda0 = im2col(da0,[8,8],'distinct');
randn('state',110);
w0 = randn(1,5120);
w0 = reshape(w0,5,1024);
alpha = 0.02;
cda1 = cda0;
for i = 1:k
    cda1(48,i) = cda0(48,i) + alpha * w0(1,i);
    cda1(55,i) = cda0(55,i) + alpha * w0(2,i);
    cda1(56,i) = cda0(56,i) + alpha * w0(3,i);
    cda1(62,i) = cda0(62,i) + alpha * w0(4,i);
    cda1(63,i) = cda0(63,i) + alpha * w0(5,i);
end
da1 = col2im(cda1,[8,8],[r,c],'distinct');
a1 = blkproc(da1,[8,8],'idct2');
figure;
subplot(1,2,1),imshow(a0,[]),title('the original image');
subplot(1,2,2),imshow(a1,[]),title('the embedded image');
dca0 = blkproc(a0,[8,8],'dct2');
dca1 = blkproc(a1,[8,8],'dct2');
cdca0 = im2col(dca0,[8,8],'distinct');
cdca1 = im2col(dca1,[8,8],'distinct');
for i = 1:k
    w1(1,i) = (cdca1(48,i) - cdca0(48,i))/alpha;
    w1(2,i) = (cdca1(55,i) - cdca0(55,i))/alpha;
```

```
w1(3,i) = (cdca1(56,i) - cdca0(56,i))/alpha;
w1(4,i) = (cdca1(62,i) - cdca0(62,i))/alpha;
w1(5,i) = (cdca1(63,i) - cdca0(63,i))/alpha;
end
SNR = sum(sum(w0.*w1))/sqrt(sum(sum(w1.^2)))
```

仿真实验结果如图 5.36 所示。

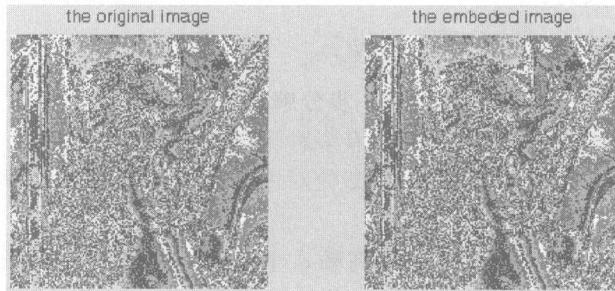


图 5.36 DCT 实验结果

## 5.6 小波域信息隐藏算法

### 5.6.1 离散小波变换 DWT

DWT 是在 1976 年 Croiser、Esteban 和 Galand 分解离散时分信号时发明的,同年 Crochier、Weber 和 Flanagan 在对语音信号进行编码时作了相同的工作。它们的分析方案称为子带编码(subband coding)。在 1983 年, Burt 定义了与子带编码很相似的技术称为金字塔编码(pyramidal coding),也称为多分辨率分析。在 1989 年, Vetterli 和 Le Gall 做了进一步实验来改进子带编码方案并且在金字塔编码方案中去除了存在的冗余。

小波分析方法是一种窗口大小固定但其形状可改变,时间窗和频率窗都可以改变的时频局域化分析方法,即在低频部分具有较高的频率分辨率和较低的时间分辨率,在高频部分具有较高的时间分辨率和较低的频率分辨率,所以被誉为数学显微镜。正是这种特性,使小波变换具有对信号的自适应性。

小波变换的含义是:把一个称为基本小波的函数  $\Psi(t)$  做位移  $\tau$  后,再在不同尺度  $\alpha$  下与待分析的信号  $X(t)$  做内积:

$$WT_x(\alpha, \tau) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{+\infty} \chi(t) \Psi^* \left( \frac{t-\tau}{\alpha} \right) dt \quad \alpha > 0$$

等效的频域表示是:

$$WT_x(\alpha, \tau) = \frac{\sqrt{\alpha}}{2\pi} \int_{-\infty}^{+\infty} X(\omega) \Psi^*(\alpha\omega) e^{j\omega\tau} d\omega, \text{ 其中 } X(\omega), \Psi(\omega) \text{ 分别是 } x(t), \Psi(t) \text{ 的傅里叶变换。}$$

小波变换对不同的频率在时域上的取样步长是可调节的,即在低频时小波变换的时间分辨率较低,而频率分辨率较高;在高频时小波变换的时间分辨率较高,而频率分辨率较

低,这符合低频信号变化缓慢而高频信号变化迅速的特点。小波变换具有以下特点和作用,具有多分辨率的特点,可以由粗到细地逐步观察信号;我们可以把小波变换看成用基本频率特性为  $\Psi(\omega)$  的带通滤波器在不同尺度  $\alpha$  下对信号做滤波;适当地选择基本小波,使  $\Psi(t)$  在时域上为有限支撑,  $\Psi(\omega)$  在频域上也比较集中,便可以使小波变换在时、频两域都具有表征信号局部特征的能力,这样就有利于检测信号的瞬态或奇异点。

离散小波变换定义为:

$$WT_f(\alpha_0^j, k\tau_0) = \int f(t) \Psi_{\alpha_0^j, k\tau_0}^* \quad j = 0, 1, 2, \dots, \quad k \in Z$$

多分辨分析只是对低频部分进行进一步分解,而高频部分则不予以考虑。

利用小波变换技术来进行隐写,目前有多种不同方案,有多阈值的小波编码方案,此种方案中将使用高值系数来隐藏秘密信息,即使对图像进行各种处理,这些系数间有一定的相关性,如果改变很大,则图像会受到影响。

小波分析是把信号分解成低频  $a_1$  和高频  $d_1$  两部分,在分解中,低频  $a_1$  中失去的信息由高频  $d_1$  捕获。在下一层的分解中,又将  $a_1$  分解成低频  $a_2$  和高频  $d_2$  两部分,低频  $a_2$  中失去的信息由高频  $d_2$  捕获,依此类推下去,可以进行更深层次的分解。

二维小波函数是通过一维小波函数经过张量积变换得到的,二维小波函数分解是把尺度  $j$  的低频部分分解成四部分:尺度  $j+1$  的低频部分和三个方向(水平、垂直、斜线)的高频部分。

DWT 是离散小波变换的缩写,离散小波变换提供了充分的用于分解和合成原始信号的信息,并且最重要的是 DWT 减少了计算时间。图像的 DWT 变换是基于 D 层的树型结构,每一层都可以使用适合的滤波器组实现。特别重要的是,对于用于提取图像采样的标准将通过滤波器组详细地描述。第一种方案,是标准小波变换,这种方案是用任何可用的小波滤波器对图像所有的行进行滤波,得到子带  $L_1$  和  $H_1$ ,然后对  $L_1$  重复,得到  $L_2$  和  $H_2$ ;这样重复  $K$  次。第二步是对图像的列进行与行相似的  $K$  次操作。如果  $K=1$ ,则分解在行与列间交替,但  $K$  可以大于 1。最终结果是在系数矩阵的左上角有一个平滑系数。这种方式与线性分解类似,分解的结果如图 5.37 所示。

LLLLLL	LLHLLL	LHLLL	HLLL
LLLLLH	LLHLLH	LHLLH	HLLH
LLLLH	LLHLH	LHLH	HLH
LLLLH	LLHH	LHH	HH

图 5.37 一种 2D-DWT 分解方式

上述的分解方式并不是标准的 DWT,所以并非常用。下面我们介绍标准的 DWT。标准的分解是将信号  $X$  通过行和列分别进行分解,重复只对低通滤波器子带图像,如图 5.38 所示。

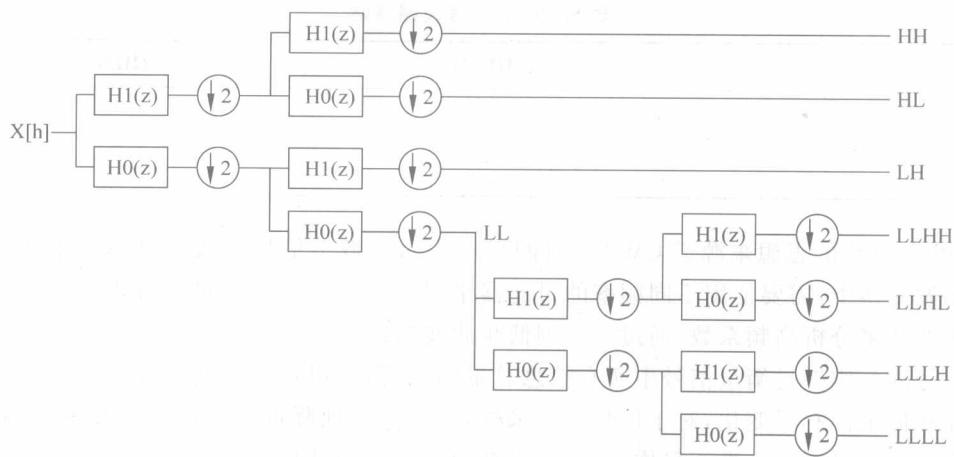


图 5.38 2D-DWT 的标准滤波器组

标准分解的示意图如图 5.39 所示。

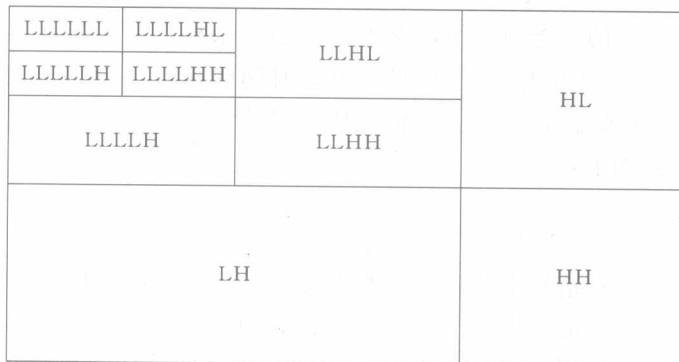


图 5.39 标准的 2D-DWT 分解

子带的命名规则如下, L 或者 H 代表使用了哪类滤波器, 单独的字线代表第一次分解的子带, 如果是 LLLL 则代表第四次分解的子带, 依此类推。

分解的过程也分为可逆的和不可逆的, 不可逆的转换通常使用在 JPEG2000 的 Daubechies 9/7 滤波器, 表 5.9 是分解滤波器的低通滤波器和高通滤波器的系数。

表 5.9 Daubechies 9/7 滤波器系数

$i$	$HL(i)$	$HH(i)$
0	0.6029490182	1.1150870524
$\pm 1$	0.2668641184	-0.5912717631
$\pm 2$	-0.0782232665	-0.0575435262
$\pm 3$	-0.0168641184	0.0912717631
$\pm 4$	0.02674875741	

可逆的变换通过用具有复数值的 5/3 滤波器, 这种滤波器的特性是在合成期间产生完全的无损的重构, 滤波器的系数如表 5.10 所示。

表 5.10 5/3 滤波器系数

$i$	$HL(i)$	$HH(i)$
0	3/4	1
$\pm 1$	1/4	-1/2
$\pm 2$	-1/8	

DWT 算法的思想来源于 CWT。时间尺度表示的数字信号由数字过滤技术获得。在离散小波变换中,需要使用不同频率的过滤器来对不同尺度的信号进行分析。通过一系列高频滤波器来分析高频系数,通过一系列低频滤波器来分析低频系数。

信号的分辨率是衡量信号中所有信息总量的尺度,可以通过过滤器加以改变。标准随高采样和低采样有所变化,对于信号的子采样,可以相应地降低信号的采样频率。例如,两个子采样在信号中隔一进行采样,采样因子减少  $n$ ,则信号中的采样次数减少  $n$  次。

高采样与低采样对应,为了增加信号的采样频率,必须对信号增加新的采样因子。例如,两个子采样是指增加一个新的采样,通常为 0 或者是在每两个信号之间的插值。高采样因子增加  $n$ ,则信号中的采样次数增加  $n$  次。

尽管这并非是唯一可能的选择,DWT 系数通常采样于 CWT 的二元栅值,如  $S_0=2$  和  $\tau_0=1$ ,则  $S=2j$  和  $\tau=k \times 2j$ ,由于信号是离散时间的函数,将用到函数和序列,序列用  $x[n]$  表示,这里的  $n$  为整数值。程序开始时使信号序列通过半通数字低通滤波器,获得  $h[n]$ 。所使用的公式如下:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k]$$

半通低通滤波器将把信号中的超过一半的最高频信号滤掉,例中,如果信号的最高的频率为 1000Hz,经过半低通滤波器后,将滤掉超过 500Hz 的频率。在此时,频率的单位特别重要,在离散信号中,信号用术语弧度表示。相应地,信号的采样频率与  $2\pi$  弧度相等,术语为径向频率。因此,在信号中的最高频率组成部分将以  $\pi$  弧度存在,如果信号以 Nyquist 的采样频率对离散信号进行采样,使用 Hz 这个单位是不恰当的。Nyquist 采样频率对应的是 rad/s。但是通常用 Hz 这个单位来进行讨论。对于离散信号的频率单位是弧度。当通过半低通滤波器之后,根据 Nyquist 原则,信号采样将滤掉一半,由于目前最高的频率是  $\pi/2$  弧度而不是  $\pi$  弧度。对于子采样则只需对信号隔一采一。半低通滤波只是过滤掉高频部分的信息,但尺度并未改变。只是子采样改变了尺度。但是子采样并没有影响分辨率,它只去掉了信号中频率的一半冗余。也就是说,低通滤波器,使分辨率减半,但尺度未变。接下来的子采样使尺度加半,但分辨率不变。使用的数学公式如下:

$$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot x[2n-k]$$

DWT 以不同的分辨率分析信号中的不同频率,它大体将信号分解为近似值和详细值。DWT 使用两个类型的函数,一种称为尺度函数,一种称为小波函数。它们各自对应相应的低通和高通滤波器。将信号分解成不同频率的带宽只需通过时间域信号的连续低通滤波器和高通滤波器,原始信号  $x[n]$  首先通过半高通滤波器  $g[n]$  和低通滤波器  $h[n]$ ,根据 Nyquist 定律,信号分解的公式如下:

$$yhigh[k] = \sum_n x[n] \cdot g[2k-n]$$

$$ylow[k] = \sum_n x[n] \cdot h[2k-n]$$

子带编码算法如图 5.40 所示,如果假设原信号  $X[n]$  有 512 个采样点,生成的频率在  $0 \sim \pi \text{ rad/s}$  之间,在第一级分解信号时,信号通过高通和低通滤波器,使用隔一采一进行子采样。高通滤波器有 256 个采样点,但是范围是  $\pi/2 \sim \pi \text{ rad/s}$ ,这 256 个采样点组成了第一级的 DWT 的系数;低通滤波器的输出也有 256 个采样点,但是范围是  $0 \sim \pi/2 \text{ rad/s}$ 。此后,信号通过相同的低通和高通滤波器,信号将进一步分解,第二次分解之后,低通滤波器的输出有 128 个采样点,频率范围是  $0 \sim \pi/4 \text{ rad/s}$ ,第二次高通滤波器的输出采样点也是 128,频率范围是  $\pi/4 \sim \pi/2 \text{ rad/s}$ ,第二次高通滤波器输出的采样点组成了第二级的 DWT 系数。依此类推,直到只有两个采样点,对于这个例子而言,则需要 8 级分解。这样,DWT 可以获得连续的系数,并且与原信号的系数数量相同。在原信号中显著的频率则在 DWT 信号范围内表现为高振幅,与 FFT 的区别在于这些频率不会丢失。但是分辨率取决于这些信号出现在哪一级,如果信号的主要信息都存在于高频信号中,那么时间的定位将更加精确。如果信号的主要信息存在于很低的频率,时间定位将不需要很精确,因为在较低频率使用较少的采样来表示这些频率。这种算法为高低频均提供了很好的时间分辨率,所以许多实际使用的信号都属于这种类型。在原始信号中频率不很显著的将具有很低的振幅,在不影响主要信息的情况下,可以丢弃部分的 DWT 系数,并且允许数据压缩。图 5.41 给出了 DWT 正常信号和数据压缩时的 DWT 信号。

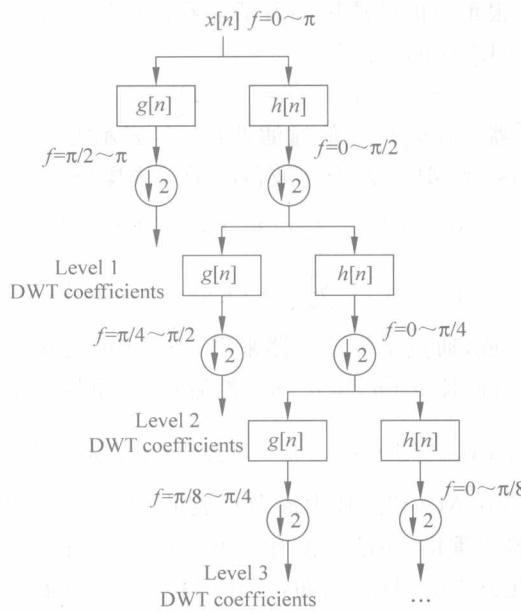


图 5.40 子带编码算法框图

横坐标是采样数值,纵坐标是规格化的振幅,图 5.41(a)给出了典型的 512 采样信号,图 5.41(b)表示了图(a)中信号的 8 级 DWT,在信号中的最后 256 次采样对应于信号中最高的频率带宽,前 128 次采样对应于第二次最高频率的带宽,依此类推。注意最先的 64 点

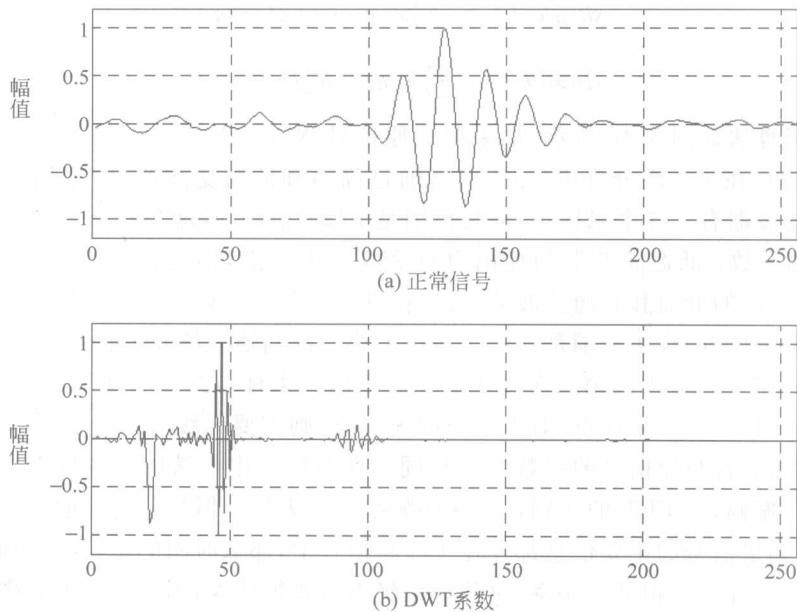


图 5.41 DWT 系数对比图

采样,它对应着较低频率的分析,携带着相关的信号的其他信息,但并不很重要,这部分的采样,就可以丢弃。这就是 DWT 能够提供很好的数据压缩方案,它能将信号的信息集中在少数的系数中。DWT 一个很重要的性质影响着低通和高通滤波器。高通滤波器和低通滤波器并不是完全独立的,它们之间的关系是:

$$g[L-1-n] = (-1)^n \cdot h[n]$$

这里, $g[n]$ 代表高通滤波器;  $h[n]$ 代表低通滤波器;  $L$  表示滤波的长度(点数)。常用的是 Quadrature Mirror Filters (QMF),两个滤波器的子采样操作可以通过下面的公式描述:

$$\begin{aligned} y_{\text{high}}[k] &= \sum_n x[n] \cdot g[-n+2k] \\ y_{\text{low}}[k] &= \sum_n x[n] \cdot h[-n+2k] \end{aligned}$$

重建的过程很容易完成,通过子带滤波器来形成标准正交基,反过来实施上述过程,对每一级采用高采样,合成过滤器  $g'[n]$  和  $h'[n]$ ,然后相加,重构的公式如下:

$$x[n] = \sum_{k=-\infty}^{\infty} (y_{\text{high}}[k] \cdot g[-n+2k]) + (y_{\text{low}}[k] \cdot h[-n+2k])$$

下面我们具体化分析在 MATLAB 中离散小波的 2D(二维)图像的实现。由于离散小波变换是基于两个滤波器的重构,小波变换有多种形式,精密采样形式的小波提供了最简洁的表示形式,但是,它有几方面的限制,例如,缺乏时移属性,在多尺度方面,不能区分方向,在图像处理中,区分方向这一点是很重要的。因此许多应用都使用扩展小波变换,扩展小波就是将  $N$  点的信号转换成  $M$  个系数,且  $M > N$ ,扩展小波有许多,因为图像大多是 2D 的,这里介绍 2D 离散小波变换。

将小波变换用于图像处理,就必须使用 2D 分解和合成滤波器组。首先使用 1D 分解滤波器对图像的列进行分解,然后再对图像的行进行分解。如果图像有  $N_1$  行和  $N_2$  列,对于

每列的 1D 滤波器将产生两个子带图像,每个图像具有  $N1/2$  行和  $N2$  列,然后再对每个子带图像进行 1D 分解,这样得到四个子带图像,2D 合成滤波器将四个子带图像合成为原图像。2D 离散小波变换将对子带图像 X 进行重复的低通滤波,离散小波变换原理图如图 5.42 所示,DWT 变换和 IDWT 变换实验结果如图 5.43 所示。

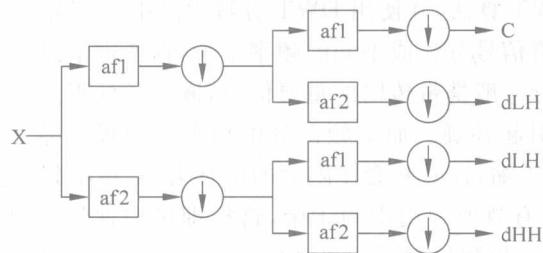


图 5.42 离散小波变换原理图

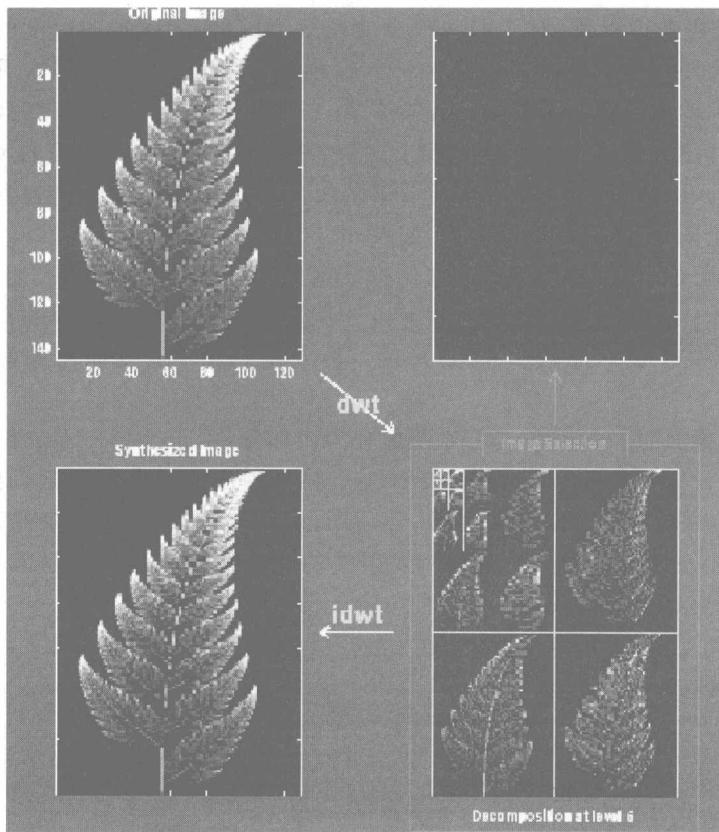


图 5.43 DWT 变换和 IDWT 变换实验结果

## 5.6.2 小波变换实现信息隐藏

在信号处理中,小波变换是非常热门的,我们将使用小波变换来进行信息隐藏。第一步

需要将图像变换到小波域,选择 HAAR 小波的原因是因为它最简单。基本过程包括低通滤波( $h[n]$ )和高通滤波( $g[n]$ ),图像以 4 种方式进行处理,求行的卷积  $h$  或  $g$ ,其结果作为行,产生 4 个输出图像,分别有  $h$  行  $h$  列,  $g$  行  $h$  列,  $h$  行  $g$  列和  $g$  行  $h$  列,使用低采样,将 4 个输出图像合成一个与原图像具有相同维度的图像。

下面,我们使用 DWT 算法,并使用 DWT 分解中的中频部分进行信息隐藏。由于具有不可比拟的优势,它能将信号分解成不同的频率子带,在考虑信息隐藏时,可以考虑高频、中频和低频,由于低频部分一般携带传输的重要信息,嵌入预处理之后的秘密信息有可能导致图像质量的下降,从而引起怀疑。而高频部分的信息一般没有意义,一般在压缩时就会舍弃,例如,常见的 JPEG 压缩,这时就会使隐藏的信息丢失,达不到信息隐藏的目的,所以,在 DWT 分解过程中,可以有效地利用中频子带,这样即可以保证信息的质量不发生变化,又能达到信息隐藏的目的。将预处理后的秘密信息嵌入到中频子带 LH 和 HL 中,这种算法的主旨思想是利用中频中的一个系数作为尺度来量化相同中频子带中的其他系数。量化步长是较大系数的固定部分。较小的系数也由这个步长进行量化。选择最大系数总数的四分之一作为重要系数进行量化,对于那些很小的系数,可以使用固定的尺度的唯一步长来量化。例如在中频子带对中,较大的系数为 100,较小的系数为 33。这里所使用的量化步长为  $1/4 \times 100 = 25$ ,因为我们嵌入的图像不是 0 就是 1,所以必须经过处理,并且要知道如何进行处理。假如此时要嵌入的数据是 0,我们选择较小的系数,乘以偶数倍的尺度来接近数据本身。如果要嵌入的数据是 1,则乘以奇数倍的尺度来接近选择的数据本身。在实际操作中,需要选定一个阈值,这里称为 stepth,如果  $\max(HL_2(i,j), LH_2(i,j)) > \text{stepth}$ ,可以使用上述的方法来进行信息隐藏,否则我们使用标准量化表进行量化后再进行隐藏,同前面讲到的 DCT 中的技术。提取方法与嵌入方法相反。我们可以通过较小系数的奇偶属性来测定给定的像素值是 0 还是 1,即较小系数/尺度。因此我们并不需要原图像就能提取隐藏的信息。

对于原始图像,使用 wavedec2 将原图像分为四部分:  $LL_1$ 、 $LH_1$ 、 $HL_1$  和  $HH_1$ 。然后对  $LL_1$  进一步分解,得到  $LL_2$ 、 $LH_2$ 、 $HL_2$  和  $HH_2$ 。将处理后的秘密信息嵌入到  $HL_2$  和  $LH_2$  对中,在这对中频中选择一个较小的系数,然后通过对较大系数  $1/n$  的奇偶变换来嵌入秘密信息。然后使用 wavedec2 将原图像和嵌入图像的所有系数组合到一起来形成伪装图像。提取图像的过程正好相反。嵌入和提取的 Matlab 编码如下:

```
clear all;
% 读入载体图像
file_name = imread('D:\MATLAB6p5\work\mylove.bmp');
cover_object = file_name;
% 读入欲隐藏图像
message = imread('D:\MATLAB6p5\work\watermark.bmp');
message = double(message);
message = fix(message./2);
message = uint8(message);
message1 = message;
% 确定载体图像的大小
Mc = size(cover_object,1);
Nc = size(cover_object,2);
% 确定欲隐藏图像大小
```

```

Mm = size(message,1);
Nm = size(message,2);
for ii = 1: Mc
for jj = 1: Nc
    watermark(ii,jj) = message(mod(ii,Mm)+1,mod(jj,Nm)+1);
end
end
watermarked_image = cover_object;
for ii = 1: Mc
for jj = 1: Nc
    watermarked_image(ii,jj) = bitset(watermarked_image(ii,jj),1,watermark(ii,jj));
end
end
imwrite(watermarked_image,'lsb_watermarked.bmp','bmp');
% 显示已伪装图像
figure(1);
imshow(watermarked_image,[]);
title('伪装图像');
figure(2);
imshow(cover_object,[]);
title('原始图像');
for ii = 1: Mc
for jj = 1: Nc
    watermark1(ii,jj) = message1(mod(ii,Mm)+1,mod(jj,Nm)+1);
end
end
figure(3);
imshow(watermark1,[]);
title('嵌入图像 1');
figure(4);
imshow(message1,[]);
title('嵌入图像 2');
提取隐藏图像
file_name = 'lsb_watermarked.bmp';
% cover_object = imread(file_name);
watermarked_image = imread(file_name); % cover_object;
Mw = size(watermarked_image,1);
Nw = size(watermarked_image,2);
for ii = 1: Mw
for jj = 1: Nw
    watermark(ii,jj) = bitget(watermarked_image(ii,jj),1);
end
end
watermark = 256 * double(watermark);
% 显示提取的隐藏图像
figure(2)
imshow(watermark,[])
title('提取的隐藏图像')

```

嵌入的图像如图 5.44 所示, 原图像和嵌入秘密信息的图像如图 5.45 所示。



图 5.44 嵌入的图像



图 5.45 原图像和嵌入秘密信息的图像

## 5.7 统计算法

在信息隐藏技术中,还有一种统计算法,最典型的统计算法是 Patchwork 算法。Patchwork 是一种使用统计技术来将秘密信息嵌入到图像的特定位的算法。用 Patchwork 算法嵌入载体图像中的秘密信息具有高斯分布的特性,基本的算法是通过选择伪随机的长度  $n$  来修改原始图像,并将嵌入的信息作为连续一对像素值。通过调节这对像素值中一方的亮度而降低另一方的亮度,这样就会得到想要的值。这个值就是嵌入的秘密信息位。我们使用 256 级线性量化系统,从 0 开始,所有的像素的亮度都很接近,所有的采样都与其他采样无关,即每个采样都是独立的。

Patchwork 和 Patchtrack 都是多层编码技术的扩充。多层编码技术就是嵌入有序顺序位的技术。多层编码技术是通过先对最强位通过 Patchwork 方案的编码来实现,通常最大的伪随机路径长度为  $n$ 。然后对其余的比特位用连续的后面的路径来表示一位。1 或 0 被编码为正或者为负(用  $S_{ni}$  表示),基本的 Patchwork 编码算法已经能进行正或负的编码(用  $\delta_i$  表示),当确定完最强位之后,只需要去识别后面数据位。正的  $S_{ni}$  代表 1,负的  $S_{ni}$  代表 0。在对数据解码时只需通过密钥计算相应的  $S_{ni}$ 。从载体图像提取每一位时,各位之间都有一定的联系。通过调整路径长度  $n$  和步长,这些统计特性可以被调整。多位编码使用差错控制编码(ECC)来调整解码时准确的  $\delta$  的相对多少。ECC 方案可以用于降低  $\delta$ ,这样也就相应降低了嵌入数据的容量。或者冗余编码也能用于确保在修改  $\delta$  时,嵌入秘密信息相关性不发生改变。ECC 编码方案下的  $S_{ni}$  如表 5.11 所示。

表 5.11 在 ECC 编码方案下  $S_{ni}$  的值

种子数	解码值	解码位	种子数	解码值	解码位
最强位(strong bit)	39896	—	4	7322	1
1	-3470	0	5	2588	1
2	-3154	0	6	5894	1
3	-13794	0			

## 5.8 图像融合算法

图像融合是将同一对象的两个或更多的图像合成在一幅图像中,以便它比原来的任何一幅更容易为人们所理解。但这两幅图像类型和大小必须一致。这一技术可应用于多频谱图像以及医学图像处理等领域,而这种技术应用到信息隐藏领域,将更好地实现信息隐藏。在图像融合技术中,可以将欲嵌入的图像嵌入到载体图像的 R、G 和 B 三个色度通道中,然后再分别提取三幅图像进行融合,能达到很好的效果。另外可以选择低频融合、高频融合、MAX、MIN、MEAN、RAND、LINEAR、UD—FUSIN、LR—FUSIN 等各种方式进行融合。

将离散小波变换与图像融合技术相结合应用于信息隐藏中,我们可以将嵌入图像直接嵌入到载体图像中,也可以利用特定的算法将两幅或两幅以上的图像融合到一起组成需要传输的秘密信息图像。融合最大的特点是:两幅或以上的图像在时空相关性以及信息各方面都能达到互补。

利用离散小波变换,将欲嵌入图像嵌入到 R、G、B 的三个色度系数上,在提取时,从 R、G、B 系数中提取嵌入图像进行融合。

嵌入过程中运用了多尺度数据融合技术,同时对载体图像和欲嵌入图像进行离散小波变换,并将嵌入图像变换后的细节系数和逼近系数分别嵌入到载体图像的各个尺度上。首先需要得到 R、G、B 三色图像。然后采用离散小波变换,并对 R、G、B 三色图像进行小波分解,得到对应的子图。同时也对原图像进行小波分解,得到相应的子图。然后进行 R、G、B 重构,将 R、G、B 图像合并得到伪装图像。下面是相应的三个分量以及最后的伪装图像,如图 5.46 所示。



图 5.46 利用融合技术生成的伪装图像

下面以两幅为  $256 \times 256$  的图像：bust 和 mask 来示例如何用小波分析进行融合。

```

Load bust
X1 = x;
Map1 = map;
Image(X1);
Load mask
X2 = X;
Map2 = map
Image(X2);
for I = 1: 256
for j = 1: 256
if(X2(I,j)>100)
X2(I,j) = 1.2 * X2(I,j);
else
X2(I,j) = 0.5 * X2(I,j);
end
end
end
[c1,s1] = wavedec2(X1,2,'sym4');
sizec1 = size(c1);
for I = 1: sizec1(2)
c1(I) = 1.2 * c1(I);
end
[c2,s2] = wavedec2(X2,2,'sym4');
c = c1 + c2;
c = 0.5 * c;
s = s1 + s2;
s = 0.5 * s
xx = waverec2(c,s,'sym4');
image(xx);

```

融合算法实验结果如图 5.47 所示。

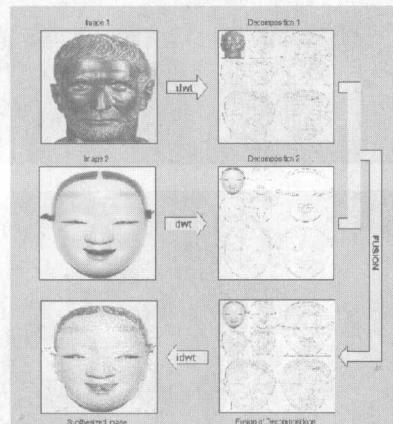


图 5.47 融合算法实验结果

## 5.9 本 章 小 结

- 对信息隐藏的算法进行了分类和概述。
- 详细分析了位平面算法、调色板算法、空间域算法、变换域算法、统计算法与图像融合算法，分析了算法的基本思想。

## 5.10 因特网资源

为跟上这个领域的发展，在因特网和 Web 上有许多可用的资源。

- <http://www.watermarkingworld.org/research.html>
- [http://www.cl.cam.ac.uk/~fapp2/watermarking/image\\_downgrading](http://www.cl.cam.ac.uk/~fapp2/watermarking/image_downgrading)
- <http://www.geog.ucs.edu/~kclarke/Corona/gallery4.htm>
- <http://www.mathworks.com>

## 5.11 复 习 题

- 5.1 位平面算法的原理？
- 5.2 什么是调色板算法原理？请在相应平台上编写相应的算法实现代码。
- 5.3 什么是空域？
- 5.4 什么是频域？
- 5.5 空域与频域有什么区别？
- 5.6 为什么目前大部分的研究都集中在小波域？小波域的优势何在？
- 5.7 如何具体实现 FFT 变换？请编写相应代码。
- 5.8 什么是 DCT 变换？如何实现 DCT 变换？
- 5.9 什么是 DWT 变换？如何实现 DWT 变换？如何在小波域中实现信息隐藏？
- 5.10 你是如何理解统计算法与图像融合算法的？

# 数字水印

### 本章目标

- 了解数字水印技术。
- 理解空域数字水印算法。
- 理解变换域算法。
- 理解可见与不可见数字水印算法。
- 理解可逆数字水印算法。
- 理解免疫数字水印算法。
- 了解多重数字水印算法。

数字水印技术是近几年来国际学术界兴起的一个前沿研究领域,特别是在网络技术迅速发展的今天,数字水印技术的研究更具有现实意义。数字水印技术的研究着重于健壮性、真伪鉴别、版权证明、网络快速自动验证、音频和视频水印等方面,其中研究最广泛的是稳健性和可验证性。数字水印的稳健性体现了水印在数字文件中的生存能力,当前大多数算法均具有一定的稳健性,但是如果同时运用各种图像攻击,那么大部分算法均会失效。如何寻找更加稳健的水印算法仍是一个急需解决的问题,同时当前水印算法在提供可靠的版权证明方面还存在一定的不完善性问题,因此提供完全的版权保护的数字水印算法也是一个重要的研究方向。在本章中,详细分析不同分类中的典型数字水印算法。

## 6.1 数字水印算法概述

数字水印有不同的分类方式,如果按水印的特性分类可以将数字水印分为鲁棒性数字水印和脆弱性数字水印;如果按水印所附载的媒体分类可以将数字水印分为图像水印、音频水印、视频水印、文本水印以及用于三维网格模型的网格水印等。如果按检测过程分类可以将数字水印分为明文水印和盲水印。如果按内容分类可以将水印划分为有意义水印和无意义水印。如果按水印隐藏的位置分类分为空域数字水印、变换域数字水印。如果按可见与否分类可以将数字水印分为可见水印和不可见水印。

无论是按哪种分类方式,应用最广泛的数字水印是用于版权保护的可见的鲁棒性水印、用于多媒体内容真实性认证的脆弱性水印,以及空域水印和变换域水印。因为无论使用哪

种数字水印,都需要嵌入过程,所以嵌入的方法一般都分为空域和变换域两种方法。在空间域方法主要有 LSB 方法、Patchwork 方法、纹理块映射编码法等。其中 LSB 方法是在像素的最不重要位嵌入水印信息,这种方法简单,但易受攻击。Patchwork 方法是将图像分成两个子集,一个子集的亮度增加,另一子集的亮度减少同样的量,这个量以不可见为标准。纹理块映射编码法是将一个基于纹理的水印嵌入到图像的具有相似纹理的部分中,这种方法是基于图像的纹理结构的,因而很难察觉水印。但是由于是嵌入图像某一部分当中,对剪切等图像处理操作鲁棒性较差,但变换域则能较好地解决这个问题。变换域的方法有 DCT、DWT、DFT 变换。变换域的方法相对于空间域方法来说,存在许多优点:

- 从提高水印的鲁棒性来看,水印应嵌入到图像在视觉上最重要的部分,而对图像来说,如果采用变换域的方法,那么图像的低频部分就能直接标记出来。
- 由于压缩算法大都在变换域进行,比如 JPEG 的 DCT、EZW 中的 DWT 等,所以可以考虑采用变换域的方法来提高抵抗压缩操作攻击的能力。
- 有些变换对某些变化有着固有的鲁棒性,例如 DFT 具有仿射不变性,对图像的空间坐标平移不敏感,因而可以利用它来恢复经过了仿射变化的图像当中的水印,又如对数极坐标变换,可以对旋转和缩放不敏感,因此利用它可以使得对水印图像的任何旋转或缩放操作都不敏感,而利用小波分析的多分辨特性,使得其对图像的剪切操作不敏感。

正是因为变换域的许多优点,变换域方法日前得以广泛的发展。

## 6.2 空域数字水印算法

在数字水印研究中,其算法主要包括空域算法,空域加性数字水印与其他域中的加性水印相同,即空间域中加性信息。水印模式与原始图像具有相同的维度,并且已经将水印模式加入到图像中。水印模式可以调制,甚至可以根据原始图像分析生成感知分析,这并不直接影响鲁棒性,视觉模型通常改进保真度以便能嵌入强水印,强水印一般具有更强的鲁棒性,其中最典型的就是最低有效位算法(LSB),我们在第 5 章中已经详细分析。另外就是变换域算法,如傅里叶变换,分块 DCT 变换和小波变换等,主要是采取扩频技术。另外还有压缩域算法,主要是基于 JPEG、MPEG 压缩过程中嵌入秘密信息的算法。另外,编码方法在数字水印中也是一个研究的执点。

### 6.2.1 最低有效位算法

数字水印最直接的嵌入方法是将水印嵌入到载体的最低有效位。虽然这种方法实现简单,但这种方法极易受到攻击,针对增加噪声或任何有损压缩攻击,都能使水印失效。另外这种方法一旦被发现,嵌入的水印极容易被未授权的第三方提取并加以调制使用。在基本 LSB 替代法的基础上改进的算法是使用伪随机发生器来确定嵌入的像素,这些位置取决于伪随机发生器的种子或密钥。这样水印的安全性提升,未授权第三方不能轻易提取水印。LSB 是信息隐藏中强有力的工具,许多广泛应用的隐写软件都采用了改进的 LSB 算法。下面介绍一下水印嵌入的其他相关技术。

除了上述嵌入方式之外,另一种水印嵌入技术是利用在载体图像中加性伪随机噪声属性。伪随机噪声为(pseudo-random noise, PN),  $W(x, y)$  是嵌入的水印,  $I(x, y)$  为载体图像, 嵌入时使用的公式如下:

$$I_w(x, y) = I(x, y) + k * W(x, y)$$

其中,  $k$  为增益因子, 随着  $k$  值的增加, 水印的鲁棒性增加, 但载体图像的质量下降。

为了提取水印, 使用相同的伪随机种子数来产生相同的伪随机噪声, 加印图像与噪声之间的相关性可以计算获得, 如果相关性超过了特定阈值  $T$ , 可以检测到水印, 如果小于特定值, 则认为此像素未嵌入水印。这种方法进行扩展就成为多位水印的嵌入, 也就是将图像分成许多块, 然后在每个块中执行相同的操作。

另外, 还可以通过许多方式来改进上述的基本算法。首先, 上述的阈值可以用逻辑 1 或 0 来确定, 1 表示水印存在, 而 0 表示水印不存在。那么上述的过程可以更方便地实现, 这要比计算加印图像到噪声间的相关性要快得多。也就是说这两种模式只需要计算其一即可。

我们还可以在图像嵌入水印之前应用预滤波, 如果能够降低载体图像与 PN 序列之间的相关性, 那么我们可以增加水印对噪声的免疫性, 通过应用边缘增加滤波, 水印的鲁棒性增强并且图像的容量不产生变化, 同时图像的质量降低很小。

最低有效位算法 LSB 是 L. F. Turner 和 R. G. van Schyndel 等人提出的一个数字水印算法, 是一种典型的空间域信息隐藏算法。LSB 算法使用特定的密钥通过 m 序列发生器产生随机信号, 然后按一定的规则排列成二维水印信号, 并逐一插入到原始图像相应像素值的最低几位。由于水印信号隐藏在最低位, 相当于叠加了一个能量微弱的信号, 因而在视觉和听觉上很难察觉。LSB 水印的检测是通过待测图像与水印图像的相关运算和统计决策实现的。Stego Dos、White Noise Storm、S-Tools 等早期数字水印算法都采用了 LSB 算法。

LSB 算法虽然可以隐藏较多的信息, 但隐藏的信息可以被轻易移去, 无法满足数字水印的鲁棒性要求, 因此现在的数字水印软件已经很少采用 LSB 算法了。不过, 作为一种大数据量的信息隐藏方法, LSB 在隐蔽通信中仍占据着相当重要的地位。

## 6.2.2 Patchwork 算法

Patchwork 是麻省理工学院媒体实验室 Walter Bander 等人提出的一种数字水印算法。Patchwork 算法首先随机选取  $N$  对像素点, 然后通过增加像素对中一个点的亮度值, 而相应降低另一个点的亮度值。这样整个图像的平均亮度保持不变。

Patchwork 方法具有伪随机性和统计特性, Patchwork 是一种使用统计技术来将秘密信息嵌入到图像的特定位。用 Patchwork 算法嵌入载体图像中的秘密信息具有高斯分布的特性, 基本的算法是通过选择伪随机的长度  $n$  来修改原始图像, 并将嵌入的信息作为连续一对像素值。通过调节这对像素值中一方的亮度而降低另一方的亮度, 这样就会得到想要的值。这个值就是嵌入的秘密信息位。我们使用 256 级线性量化系统, 从 0 开始, 所有的像素的亮度都很接近, 所有的采样都与其他采样无关, 即每个采样都是独立的。Patchwork 算法处理过程如下:

提高  $a_i$  点的亮度, 典型的范围是 256 中的 1~5。降低  $b_i$  同样的亮度, 不断地重复上述

过程,典型的是 10000 次,就已经将秘密信息嵌入到了载体图像中。在 Patchwork 算法中有许多限制,首先就是嵌入的秘密信息容量有限,所以比较适用于数字水印这个分支,另外,如果没有第一步的处理,也就是随机种子数的选择这一步,将很容易定位到嵌入信息的位置,但很难提取出秘密信息,所以经过预处理,则使 Patchwork 算法安全性更高一层。这种算法很易受到攻击,如果图像进行平均,嵌入部分就会比平均部分过亮或者过暗,这是这种算法的弱点。

使用特定的密钥来产生随机种子数来选择  $(a_i, b_i)$ ,这一步很重要,因为编码器需要在解码过程中使用这一点。

在采样中任取两点,这里命名为 A 和 B,在载体图像中随机选择 a 和 b,让 a 的亮度与 A 相同,b 的亮度与 B 相同。这里,假定  $S=a-b$ 。S 最佳的值是 0。当经过多次的重复处理过程,最终 S 的最佳平均值也为 0。但 S 的值是变化的,它的值在无限逼近最佳值。

Patchwork 数字水印隐藏在特定图像区域的统计特性中,其鲁棒性很强,可以有效地抵抗剪切、灰度校正、有损压缩等攻击,适当地调整参数,Patchwork 方法对 JPEG 压缩、FIR 滤波以及图像裁剪有一定的抵抗力。其缺陷是数据量较低,对仿射变换敏感,对多拷贝平均攻击的抵抗力较弱。

Patchwork 算法嵌入的是一种数据量较小、能见度很低、鲁棒性很强的数字水印,能够抗图像剪裁、模糊化和色彩抖动。“Patchwork”一词原指一种用各种颜色和形状的碎布片拼接而成的布料 v,它形象地说明了该算法的核心思想,即在图像域上通过大量的模式冗余来实现鲁棒数字水印。与大多数图像域数字水印算法不同,Patchwork 并不是将水印隐藏在图像数据的最低有效位 LSB 中,而是隐藏在图像数据的统计特性中。

以隐藏 1 位数据为例,Patchwork 算法首先通过密钥产生两个随机数据序列,分别按图像的尺寸进行缩放,成为随机点坐标序列。然后将其中一个坐标序列对应的像素亮度值降低,同时升高另一坐标序列对应的像素亮度。由于亮度变化的幅度很小,而且随机散布,并不集中,所以不会明显影响图像质量。为了提高鲁棒性,还可以改变随机点邻域中的像素亮度,这样就形成了图像域上亮、暗模式的铺砌。

影响 Patchwork 算法使用效果的因素很多,主要有以下几方面。

(1) Patch 的深度。Patch 的深度是指对随机点邻域灰度值改变的幅度,深度越大,水印的鲁棒性越强,但同时也会影响隐蔽性,提高能见度。

(2) Patch 的尺寸。大尺寸的 Patch 可以更好地抗旋转、位移等操作,但尺寸的增大必然会引起水印信息量的减少,造成 Patch 相互重叠。具体应用时必须在 Patch 的尺寸和数量两者之间进行折衷。

(3) Patch 的轮廓。具有陡峭边缘的 Patch 会增加图像的高频能量,虽然这有利于水印的隐藏,但也使水印容易被有损压缩所破坏。相反,具有平滑边缘的 Patch 可以很好地抗有损压缩,但易于引起视觉注意。合理的解决方案应该是在考虑到可能会遭受的攻击后确定,如果面临有损压缩的攻击,则应采用具有平滑边缘的 Patch,使水印能量集中于低频;反之,如果面临对比度调整的攻击,则应采用具有陡峭边缘的 Patch,使水印能量集中于高频。如果对所面临的攻击没有准确的估计,则应使水印的能量散布于整个频谱。

(4) Patch 的排列。Patch 的排列应尽量不形成明显的边界,因为人眼对灰度边界十分敏感,W. Bender 建议采用随机的六角形排列。

(5) Patch 的数量。Patch 的数量越多,解码越可靠,但这同时也会牺牲图像的质量。除了这些因素之外,还可以在 Patchwork 水印算法中融合许多图像滤波技术,如采用视觉掩模技术等,来提高水印的隐蔽性或鲁棒性。

Patch Track 是与 Patchwork 算法不同的算法,它重点在于信息隐藏的内在的问题。它用于数据的提取队列。这个问题可以通过搜索方法,包括粗略定位检测定位法,随机搜索法和梯度下降搜索。

水印解码程序 Patch Track 实际上是一个统计信号检测器。Patch Track 首先对扫描后的票据图像进行矫正处理,克服由旋转、破损等带来的水印特性变化。随后,Patch Track 使用解密密钥产生二维随机点坐标序列,形成解码窗口。通过构造适当的像素灰度统计量,可以判断解码窗口中是否包含有 Patchwork 水印。

为了实现打印机的自动票据识别与票据拒打功能,麻省理工学院数据隐藏研究小组提出了线状数字水印——数字隐线(Tartan Thread)技术。与隐蔽标识方法不同,Tartan Thread 是一种主动防护技术,它必须与票据制作者配合,在真实的票据图案中加入防伪水印,这种线状的数字水印能够存在于扫描后的票据图像中,在打印输出时,打印机驱动程序中的水印解码模块能快速解读水印,一旦发现票据防伪隐线,就立即拒绝打印输出。

数字隐线防伪方案面临的最大难点是解码空间的问题。一般来说,打印机驱动程序只缓存几行像素,在打印过程中,内存中自始至终没有一个完整的打印图像,所以数字隐线的解码空间十分狭小。另外,数字隐线的解读过程必须非常迅速,如果过多地影响打印效率,则无论是打印机厂商还是用户都难以接受。

Tartan Thread 数字隐线的核心技术是一维扩频调制,即将水印信息用扩频码调制成具有噪声性质的信号,叠加在票据图像上。解码器使用同样的扩频码通过解扩读取数字隐线。

Patchwork 方法安全性很好,但嵌入信息的容量受到了限制,另外 Burgett 等人提出了脉冲嵌入系统,在这种方案中,位置序列用于映射像素的位置,然后对此序列编码,脉冲编码是有层次的。在空域中的水印的缺点就是鲁棒性较差,很容易受到破坏。

### 6.3 变换域算法

空域中的数字水印不同程度会造成图像质量的下降,根据人类视觉特性,在变换域能更好地调整水印而不影响载体图像的质量。最广泛研究的 DCT、DWT 域都是能量保留,正交变换的结果。高维媒介空间中的每条轴线都对应着工作空间相应的值,如在图像空间中,轴线上的每个值对应着一个像素的亮度,在音频空间,每个轴线可能对应着音频采样。因此在这个空间中的每个点都在工作空间中有所对应。当应用能量保留,正交变换时,我们只是旋转相应的系统,使每条轴线有新的表示,如果以前代表像素,此时可能代表频率。如果水印算法与坐标轴无关,应用任何一种变换对其性能都没有任何影响。例如,考虑非常简单的应用变换的嵌入,通过增强白噪声的方式,由于白噪声具有放射性和对称性,水印模式的分布

概率与坐标无关。这就意味着,不论使用任何变换,都不会改变水印图像的分布。也就是说变换对其性能没有任何影响。如果是非线性嵌入的水印,例如,FFT 的幅值嵌入,或者应用某种形式的感知建模(感知建模一般都是非线性的),变换只是作为嵌入的一种工具,即使图像变换到频域,使用频域的某些特性来完成数字水印的嵌入。

目前广泛使用的变换域为 DCT 和 DWT,下面分别介绍。

### 6.3.1 DCT 算法

DCT 变换域数字水印是目前研究最多的一种数字水印,它具有鲁棒性强、隐蔽性好的特点。其主要思想是水印信号应该嵌入在频谱空间源数据中人类感觉最重要的部分,这种重要部分就是低频分量。在图像的 DCT 变换域上选择中低频系数叠加水印信息。之所以选择中低频系数,是因为人眼的感觉主要集中在这一频段,攻击者在破坏水印的过程中,不可避免地会引起图像质量的严重下降,一般的图像处理过程也不会改变这部分数据。水印信号应该由具有高斯分布的独立同分布随机实数序列构成。这使得水印经受多拷贝联合攻击的能力大大增强。

实现方法是:首先以密钥为种子来产生伪随机序列,该序列具有高斯  $N(0,1)$  分布,密钥一般由作者的标识码和图像的哈希值组成,对整幅图像做 DCT 变换,用伪随机高斯序列来调制(叠加)该图像除直流分量(DC)外的 1000 个最大的 DCT 系数。

由于 JPEG、MPEG 等压缩算法的核心是在 DCT 变换域上进行数据量化,所以通过巧妙地融合水印过程与量化过程,就可以使水印抵御有损压缩。此外,DCT 变换域系数的统计分布有比较好的数学模型,可以从理论上估计水印的信息量。

因为早期的 DCT 集中在低频系数上,现在我们分析一下中频的 DCT 系数,通过数字水印嵌入公式可知, $k$  不仅可以作为增益因子,而且作为阈值更好。在前面的技术中,通过增加  $k$  来增强水印的鲁棒性,但是降低了载体图像的质量。而使用中频系数时,我们计算两个系数的幅值差,如果两者差不超过  $k$  时,这对系数才能满足我们的需求。并且不降低载体图像的质量。同时 DCT 变换一般采用  $8 \times 8$  的块,其目的是抵御 JPEG 压缩攻击,块越大将获得更好的结果,但以牺牲信息容量为代价。同时使用中频系数可以更好地抵御高斯噪声攻击与 JPEG 攻击。

另外 Kankanhalli 等已经提出了基于图像纹理的 DCT 域可见数字水印技术,纹理块映射是将水印信息隐藏在图像的随机纹理区域中,利用纹理间的相似性掩盖水印信息。该算法对滤波、压缩和扭转等操作具有抵抗能力,但需要人工干预。算法的公式如下:

$$c'_{ij}(n) = \alpha_n c_{ij}(n) + \beta_n w_{ij}(n) \quad n = 1, 2, \dots$$

$$w_{ij}(n) = \alpha_n c_{ij}(n) + \beta_n w_{ij}(n) \quad n = 1, 2, \dots$$

首先将图像分成不同的块,通过感知的方法对分成的块进行分类,并且调制原图像块  $n$  的系数  $\alpha_n$  和  $\beta_n$ ,其中  $\alpha_n$  是尺度因子,  $\beta_n$  是嵌入因子。 $c_{ij}(n)$  是原图像块的 DCT 系数, $w_{ij}(n)$  是嵌入水印后图像块的系数。使用 HVS(human visual system)的纹理敏感度  $\alpha_n$  和  $\beta_n$  能使嵌入水印图像具有更佳的感知质量。我们称前者是尺度因子,后者为嵌入因子。通过调制不同的值来使可见水印具有更高的鲁棒性。找到最佳的尺度因子  $\alpha_n$  和嵌入因子  $\beta_n$ ,是保证嵌入水印后的图像保真度不降低的重要步骤。因为边缘块是反映图像的轮廓,所以尽量在使用边缘块来完成水印的嵌入时,尺度因子  $\alpha_n$  逼近最大尺度因子  $\alpha_{\max}$ ,而嵌入因子应

逼近最小嵌入因子  $\beta_{\min}$ 。当背景具有强纹理时失真度最小。在高纹理的块中,能量均匀分布在 AC 的 DCT 系数上。这就意味着高纹理块 AC 的 DCT 系数变化很小,我们可以在这些块中嵌入较多的水印。因此为了方便,我们假设  $\alpha_n$  与变量  $\sigma_n$  成正比,  $\beta_n$  与变量  $\sigma_n$  成反比,每个图像块的平均灰度值为  $\mu_n$ 。中强度值为  $\mu_n = \mu$  的块比其他的块对噪声更敏感。这就说明当  $\mu_n < \mu$  时,  $\alpha_n$  的值随  $\mu_n$  增大, 反之降低。 $\alpha_n$  和  $\mu_n$  的关系是变形的高斯分布。

Ingemar 等人提出了安全扩频水印(Secure Spread Spectrum Watermarking),他们将水印信息分散到数据的最重要光谱分量上。算法的描述如下:

- (1) 生成数字水印:  $X = x_1, x_2, \dots, x_n, x_i$  是 Gaussian 随机变量  $N(0, 1)$ 。
- (2) 将水印插入到序列  $V = v_1, v_2, \dots, v_n$ , 得到调整后的序列值  $V' = v'_1, v'_2, \dots, v'_n$ 。
- (3) 计算原始图像的 DCT 变换, 将水印嵌入到感知重要的区域。
- 水印的提取算法:
  - (1) 计算水印图像的 DCT。
  - (2) 计算原图像的 DCT。
  - (3) 水印图像 DCT 减去原图像 DCT 得到可能的水印  $X^*$ 。
  - (4) 计算提取水印与真实水印之间的相似度( $\cdot$  代表  $X^*$  和  $X$  的标量积或数量积):

$$\text{sim}(X, X^*) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X}}$$

ZHAO 等人提出了另外一种方法, 将图像分成  $8 \times 8$  块, 然后计算每个块的 DCT 系数, 选择两个 DCT 系数, 计算两者之间的关系, 然后嵌入水印, 同时选择的系数要基于 JPEG 量化表, 这样在反量化时才能重现带水印的系数。图像分成  $8 \times 8$  RGB 块, 然后将这些块转换成 YCC 分量, 我们使用亮度值进行操作, 执行 DCT 变换, 根据量化表选择两个系数来表示亮度分量, 亮度值用于水印的编码。水印的编码算法如下:

$a$  是第一个系数,  $b$  是第二个系数。两个系数之间的改变与否依赖于水印的数据位。Koch 等人提出, 图像分成  $8 \times 8$  RGB 块, 计算每个块的 DCT, 然后量化, 量化后具有三种频率, 这里取中频范围进行调制, 使其相对的强度编码是 1 或 0。选择  $a, b, c$  三个系数。计算最大值和最小值之间的差分, 如果差分大于常量, MD 块标记为无效, 然后调整系数直至满足条件。算法框图如图 6.1 所示。

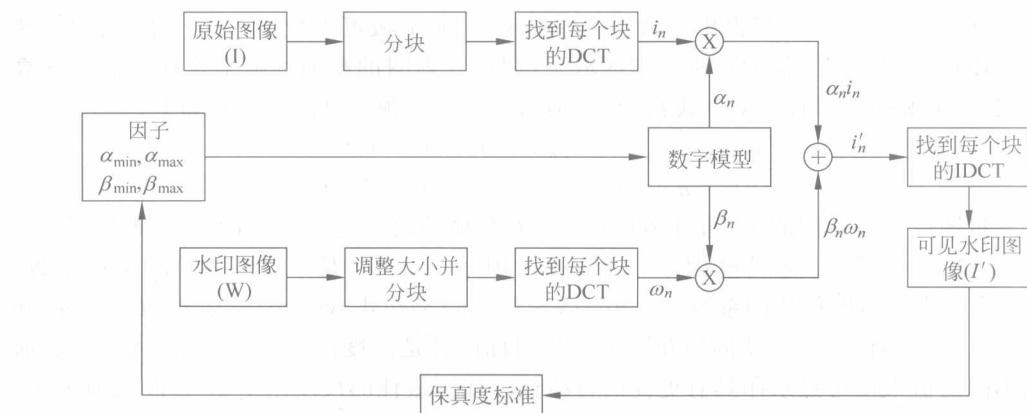


图 6.1 算法框图

本节将介绍 DWT 算法在数字水印中的应用，包括如何利用 DWT 算法生成不可见水印。

### 6.3.2 DWT 算法

DWT 算法在第 5 章中已介绍，这里主要针对数字水印中 DWT 算法的应用。下面介绍基于小波变换的不可见水印。

不可见水印具有较强的隐蔽性，但由于含印载体会受到常规的处理和有意的攻击，因此要求不可见水印必须能够经受这些处理和攻击。用于版权认证的不可见水印必须具有较强的鲁棒性、安全性和透明性。要达到透明性要求，不可见水印必须低强度嵌入；为了提高水印的鲁棒性，我们将水印嵌入图像的重要小波树中；为增强水印安全性，我们将水印进行混沌置乱。具体步骤如下。

(1) 对图像进行多分辨率小波变换，生成图像的重要小波树。小波变换用于图像分析的基本思想就是把图像进行多分辨率分解，将图像分解成不同空间，不同频率的子图像。图像经过小波变换后被分割成 4 个频带：低频、水平、垂直和对角线，低频部分还可以继续分解。在以小波分解的图像数据中，可以构成若干个四叉树，粗糙信息层( $HL_3, LH_3, HH_3$ )中的小波系数是其下一个精细层( $HL_2, LH_2, HH_2$ )中 4 个对应位置小波系数的父结点，它代表了精细层次中对应位置小波系数幅度的加权平均值。这 4 个对应位置的小波系数则称为子结点，父结点与子结点在各自分解层上的重要性较相似，且低频小波系数通常要远大于其相对应的高频小波系数。这样小波树把空间域同一位置、不同尺度、不同方向的小波系数组织在一起。图像小波分解树如图 6.2 所示。

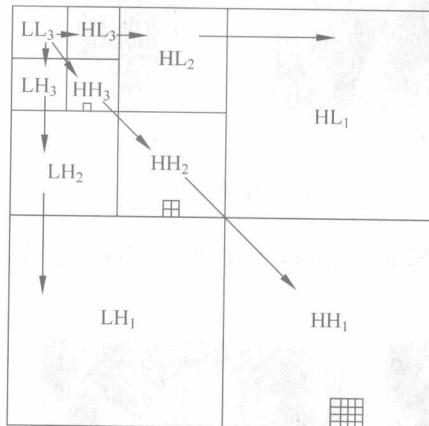


图 6.2 图像小波分解树

将低频逼近子带排除在水印嵌入区域之外，因为该子带的修改极易引起视觉上的察觉。为了提高水印的鲁棒性，将码组矩阵嵌入到重要小波树中。在最低频的  $LL_3$  细节子带中设定阈值  $T$ ，将高于阈值系数的小波树确定为重要小波树。

(2) 对原始水印图像进行混沌置乱。对图像进行多次二维 Arnold 置乱，置乱次数作为密钥的一部分。Arnold 变换是一种混沌变换，能提高水印的保密性，并且能打乱水印信号的自身相关性。

(3) 通过计算邻域相关性将水印嵌入重要小波树系数中。为了达到鲁棒性和盲提取两项要求，对重要小波树系数进行  $2 \times 2$  分块，通过计算分块左上顶点的块内邻域相关性进行

水印嵌入。对于第  $K$  个待嵌入分块, 定义左上顶点为嵌入点, 其他 3 点为邻点。定义邻域系数平均值

$$\text{ave}_k = \frac{1}{3} \left[ \sum_{i=1}^2 \sum_{j=1}^2 x_k(i, j) - x_k(1, 1) \right]$$

嵌入后小波系数为

$$x'_k(1, 1) = \begin{cases} \max v_k(1 + \alpha), & x_k(1, 1), w_k = 1 \\ \min v_k(1 - \alpha), & x_k(1, 1), w_k = 0 \end{cases}$$

其中,  $\alpha$  为嵌入深度因子;  $\alpha$  越大鲁棒性越高, 对载体图像的质量影响越大。提取规则为

$$w_k = \begin{cases} 1, & x_k - v_k > 0 \\ 0, & x_k - v_k < 0 \end{cases}$$

(4) 将修改后的子带进行逆小波变换, 得到含水印图像。

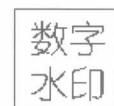
在仿真实验中, 原始图像采用  $512 \times 512$  的灰度图像, 可见水印图像采用  $123 \times 152$  的徽标, 不可见数字水印图像采用  $64 \times 64$  二值图像, 如图 6.3 所示。



(a) Lena 图像



(b) 可见水印



(c) 不可见水印

图 6.3 原始图像和水印

双重水印嵌入提取图如图 6.4 所示, 篡改、剪裁与噪声攻击及提取的不可见水印分别如图 6.5~图 6.7 所示。



(a) 含可见水印图像



(b) 含双重水印图像



(c) 提取的不可见水印

图 6.4 双重水印嵌入提取效果图,  $\alpha=0.015$

由于采用图像融合技术, 可见水印很难通过计算从图像中去除。攻击者只能通过蛮力擦除可见水印, 仍然可以可靠地提取不可见水印来证明作品的版权。上面的例子是遭到噪

声攻击时含水印图像及提取的水印,其中的鲁棒性不可见水印有较强的抗水印攻击能力。

上述实验结果表明,使用 DWT 技术的双重数字水印,使得非法用户难以利用作品内容;即使在蛮力擦除可见水印后依然能够通过不可见水印证明作品版权。



(a) 遭篡改含水印图像 (b) 提取的不可见水印

图 6.5 篡改攻击及提取的不可见水印



(a) 遭裁剪含水印图像 (b) 提取的不可见水印

图 6.6 裁剪攻击及提取的不可见水印



(a) 遭噪声含水印图像 (b) 提取的不可见水印

图 6.7 噪声攻击及提取的不可见水印

## 6.4 可见与不可见数字水印算法

数字水印分为可见水印与不可见水印。目前绝大多数文献集中于不可见水印的研究和实现上,可见水印的相关文献很少。不可见水印较之可见水印具有不可感知性,非常适合各

种媒体的版权认证、内容完整性保护和篡改认证。对作品的非法传播有一定威慑作用；但由于嵌入不可见水印后媒体质量较高，不会影响非法用户的“享用”，难以阻止非法用户对内容的非法利用。

可见数字水印将水印图融合到作品中，可以掩盖部分图像数据，可以减少非法窃取者窃取含有可见水印产品的意图，也可以起标注或宣传作用，能够防止非法用户获取部分重要信息。但在一定程度上破坏了原图像，降低了嵌入数字水印后产品的商业价值，且不适用于音频等非视觉媒体。可见数字水印主要用于图像和视频的版权保护，在网上数字图书馆、网络电视越来越多的今天，实用的可见水印的方法尤有意义。

可见水印是一种主动的保护方式，但也更容易成为攻击目标。不可见数字水印可以在可见数字水印完全被擦除的情况下保护图像。可见数字水印，主要用于当场声明对产品的所有权、著作权及来源，起到一个宣传广告或约束的作用。可感知水印一般为较淡的或半透明的不碍观瞻的图案；例如，电视台节目播放的同时，在某个角落插上电视台的半透明标志。另一个用途是为了在线分发作品，比如先将一个低分辨率的有可见水印的图像免费送人，其水印往往是拥有者或卖主的信息，它提供了寻找原高分辨率作品的线索，若想得到高分辨率的原作品则需付费。有些公司在产品出售前为了在网络上宣传其产品，先做上可逆可见水印分发，付费购买时，再用专用软件将可见水印去掉，加入不可见水印（发行人、分发商、最终用户等的信息）。可见水印还有另一些用途，那就是为了节约带宽、存储空间等原因，在 VCD、DVD 等电影拷贝中用嵌入不可见水印的方式配上多种语言的副标题和字幕，待播放时由硬件根据需要实时地解出每一帧中的水印文字，将其显示在屏幕上。

可见水印在某些产品中或多或少降低了作品的观赏价值，使其用途相对受到一定限制。不易感知的水印的应用层次更高，制作难度更大。可见数字水印是常见的一种水印，经常用于保护公开的可用的图像。在可见数字水印中更强调针对各种攻击的鲁棒性，即使攻击者发现了隐藏信息的存在，但他也很难破坏嵌入的水印。可见水印最常见的例子是在图版或视频上的商标，用于让观众知道版权所有者。通常可见水印可以通过空域或变换域的隐写算法来完成，可以通过改变图像的亮度来嵌入水印，也可以在 DCT 域或 DWT 域嵌入水印。

对于可见水印一般有如下要求：

- 无论在单色或彩色图像中，可见数字水印都应该非常明显。
- 为了防止通过剪切破坏水印，应该将可见水印分布在图像的最重要部分。
- 可见水印不应该影响在水印之下的重要的图像细节内容。
- 可见水印应该是很难移去的，移去水印很费劲或要付出的代价远远超出直接购买此图像。
- 加入水印应该自动完成。

在当前的文献中，可用的可见水印技术很少。在国外的一些数字图书馆中使用了可见数字水印技术来标识著名画家的作品。

下面介绍一个可见水印与不可见水印结合应用的算法。

结合可见水印和不可见水印各自的优点，二者结合起来可以更加有效地保护作品。两者的嵌入有两种先后顺序：①先嵌入可见水印，再嵌入不可见水印；②先嵌入不可见水印，再嵌入可见水印。但相比而言，保证不可见水印的完整性更重要，且其对可见水印的影响相对较小。因此本文采用先行嵌入鲁棒的可见水印，再迭代嵌入鲁棒的不可见水印的双重水

印算法,如图 6.8 所示。

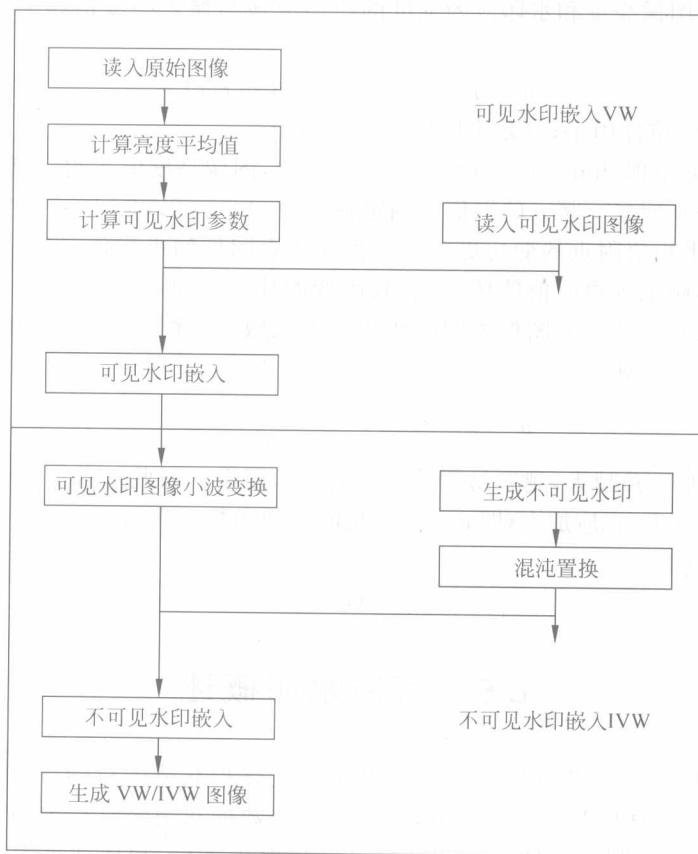


图 6.8 可见与不可见算法框图

算法主要步骤包括:

- (1) 读入原始图像并计算其亮度平均值。
- (2) 计算可见水印嵌入的拉伸因子等参数。
- (3) 读入可见水印图像。
- (4) 根据拉伸因子进行可见水印嵌入,生成含可见水印图像。
- (5) 对含可见水印图像进行小波变换。
- (6) 生成不可见水印,并对其进行混沌置乱。
- (7) 在含水印图像重要小波树系数中根据邻域相关性嵌入不可见水印。
- (8) 逆小波变换,得到还原图像。

另外目前通常将图像融合技术应用于可见数字水印中,下面加以介绍。

可见水印处理利用主图像局部特征信息指导水印嵌入,以获得可见性好、不突出、难去除的水印。可见水印有以下要求:①可见水印应能保护作品重要内容。这表明水印应该覆盖图像重要区域,并且应在所覆盖的区域可见;②水印一般应在所覆盖的区域上半透明可见,不完全破坏细节。因此嵌入程度应随主图像局部特征的变化而变化,通常是足够突出地显示以阻止非法使用;③非法去除水印难度很大。

要达到上述要求,需要同时调整图像亮度和水印的强度,在空间域逐像素修改图像。要保证水印透明,对图像亮度和水印嵌入强度的拉伸应该与像素亮度值相关。可取嵌入公式如下:

$$h(i,j) = \alpha h(i,j) + \beta w(i,j)$$

其中,  $\alpha$  是图像亮度拉伸因子;  $\beta$  是水印嵌入深度因子。

根据人类视觉原理(human visual system, HVS),图像亮度值大约在缩小到 0.9 倍的情况下,图像失真并不明显。将  $\alpha$  的取值范围定在区间[0.85, 0.99]上。另外,人的视觉系统对接近图像亮度平均值附近的变化更敏感。若  $\mu$  代表图像的平均值,它的细小改变人眼很容易发觉。反之,越是远离此值的像素点,其改变的比例  $\alpha$  应越大,即  $\alpha$  是  $h(i,j) - \mu$  的增函数。同时,人眼对有背景的图像值的改变往往不太敏感,背景越深越不敏感,则  $\alpha$  是  $\mu$  的反函数。综上,可取  $\alpha$  如下:

$$\alpha = 0.8 + 0.2 \frac{(h(i,j) - u)^2}{u^2}$$

图像的亮度  $h(i,j)$  越大,越容易覆盖背后的水印图案。因此随着图像的亮度加大,  $\beta$  的取值(水印的嵌入深度)也应加大,即  $\beta$  是图像亮度的增函数。常用的函数有:  $y=x_m$  ( $m>0$ ),  $y=a_x$  ( $a>1$ ) 等,可取

$$\beta = 0.6 \exp(h(i,j)/255)$$

## 6.5 可逆水印概述

数字水印将版权等信息隐藏到载体中,从而达到保护载体的目的。数字水印连接着两组数据: 隐体数据和载体数据。在多数情况下,载体数据在隐藏过程中会出现失真而无法恢复到原始状态。这就是说,即使隐体数据已经提取出来后,载体数据也遭到了永久性失真。但在某些应用中,对数据像素的极小更改都是不允许的。例如军事遥感图像、医学图像、高能物理图像、法律证据等敏感图像。在这些应用中,任何一个像素的信息都认为是非常重要的。对数据的任何修改都会影响图像的可信度。这些应用均要求使用最原始的图像数据。

显然,大多数现有的数字水印技术都不是可逆的。例如广泛应用的扩频信息隐藏方法在取整时出现了截断或者舍入误差; 最低有效位 LSB(或广义的 LSB 算法)算法,由于位替换而对原始信息失去“记忆”; 基于量化索引调制的水印会出现量化误差。这些算法都因对原始信息或嵌入过程无完整记忆,而无法纠正失真来进行恢复。这就需要能将载体数据恢复到原始状态的数字水印技术。

能将载体数据恢复到原始状态或非常接近原始状态的水印技术,称之为可逆数字水印技术(reversible digital watermarking)。可逆数字水印技术也可称之为可恢复(reversible)、无损(lossless)、无失真(distortion-Free)或者可逆(invertible)数字水印/信息隐藏技术等。可逆数字水印在提取出隐体数据后,利用隐体数据来无损恢复载体数据,在隐体数据和载体数据之间建立了巧妙的关系。

可逆数字水印可以用于数字产品的推广。比如付费视频应用中,用户在确定购买正式视频之前,需要对视频内容大致了解,以决定是否正式付费购买; 商家需要借此机会向用户展示内容吸引顾客,但商家又不愿将清晰作品送给用户。在这种两难情况下,可采用可逆数

字水印技术,对作品进行加上水印使顾客仅能“浅尝”。用户付费后,可经过授权后将视频内容恢复到高清(high definition)状态。

根据水印的恢复能力,我们将可逆数字水印分为强可逆数字水印和弱可逆数字水印。强可逆数字水印中,恢复后的载体与原始载体完全一致,没有任何差别;弱可逆数字水印可将含印载体恢复到几乎与原始载体完全一致的状态。用公式表示如下:

假设原始载体为  $x \in X$ ,含印载体为  $y \in X$ ,恢复后的载体为  $x' \in X$ ,则

- (1) 强可逆数字水印满足  $x' = x$ ,即  $\|x' - x\|_x = 0$ ;
- (2) 弱可逆数字水印满足  $x' \approx x$ ,即  $\|x' - x\|_x < \epsilon$  且  $\|x' - x\|_x \ll \|y - x\|_x$ , $\epsilon$  为很小的正数。

上述定义中,  $\|\cdot\|_x$  是载体空间元素的范数。

### 6.5.1 可逆数字水印现有算法

可逆水印的实现方法,通常当嵌入空间确定后,将该空间内的原始数据状态进行无损压缩,从而空出部分空间能够进行水印数据嵌入。这就要求选定的空间其状态数据冗余度较高,能够进行有效压缩,从而空出更多的空间用于隐藏数据嵌入。

通常的高容可逆水印方法是选定隐藏嵌入区域(如某些像素的最低位),然后将被覆盖数据(用来恢复该区域)和纯载荷数据一同嵌入该区域,如图 6.9 所示。如果要嵌入的信息量(载荷数据和用于恢复的嵌入区域原始数据值)高于嵌入区域容量,算法就依赖于嵌入区域原始数据的无损压缩,留出的空间用于嵌入载荷数据。

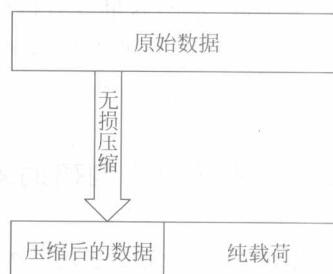


图 6.9 可逆数字水印实现原理

最早的可逆数字水印算法是在 2000 年 Barton 的专利中出现的。其基本思路是将 JPEG 等载体待覆盖的位进行压缩,留出空间填上认证信息。Honsinger 的可逆嵌入算法是从含印图像中重构被覆盖的数据以无损恢复原始图像。Macq 提出采用扩展的 patchwork 算法实现可逆隐藏,但含印载体的椒盐噪声较为严重。Fridrich 等人提出了基于块状态编码的可逆隐藏算法,其方法是对图像分块并将信息隐藏在块的状态中。De Vleeschouwer 提出了一种基于双射函数的、可循环解释的可恢复数据隐藏算法。Kalker 等人提出了基于无损数据压缩的可逆数据隐藏的理论容量极限,并给出了可操作性的码的构造。Celik 等人提出了基于压缩量化余数的可逆数据隐藏算法,采用的是 CALIC 压缩算法,将量化值作为边信息以实现对量化余数的更高压缩率,从而得到高的嵌入容量。Xuan 提出了整型小波变换的无失真数据隐藏算法,隐藏容量较高。

2003 年 Tian 提出了一种迄今为止容量最高的一类可逆数字水印算法——基于扩大差值的高容量无损数字水印算法,方法是选定相邻的像素对计算其整数均值和差值,在差值的最低位嵌入比特  $b$ ,即  $h' = 2 \times h + b$ 。该方法需要对像素对进行分类处理。像素对根据能否嵌入和能否更改的性质差异,分为四类: EZ(包括所有可扩展像素对且  $h=0, h=-1$ )、EN(包括所有可扩展像素对且  $h \notin EZ$ )、CN(包括所有可更改类,且  $h \notin (EZ \cup EN)$ )、NC(所有不可更改的  $h$  值)。对不同像素对采用不同的处理。但该算法有两个严重缺陷:

(1) 嵌入和提取过程处理不对称,算法实现繁杂,计算量大。

(2) 需要同时存储不可嵌入但可更改类的像素最低位,并对这些像素对进行最低位替换,这种替换不会带来任何容量增加,但却降低了载体图像质量。特别是在容载要求较低的情况下,仍然要对所有可更改类进行最低位替换,给含印图像质量带来了不必要的损失,并在嵌入和提取时大大增加了算法处理的复杂度。所以 Tian 算法的含印图像质量偏低(一般在 40dB 以下,最高不超过 44.2dB)。

Alattar 在 Tian 算法的基础上,将“像素对”扩展为三像素组[XXX]、四像素组[XXXX]、像素对向量[XXX]的方法,以能在每像素组中嵌入多个比特,从而进一步增加隐藏容量。这种改进在相同的含印载体保真度下进行对比,在保真度较低时,容量增益比较明显;但在保真度要求较高时,容量反而有较明显的下降。

2006 年 Ni 提出了一种基于柱状图调整的高保真度可逆数据隐藏算法。算法通过微调图像灰度柱状图的零点或最低点处的像素点的灰度值,将数据嵌入到图像中。在  $512 \times 512 \times 8$  的载体灰度图像中,嵌入 5~80KB 的数据后,含印图像质量能保持在 48dB 以上。其最低保真度高于目前现有大部分可逆水印算法。但其缺陷是最大容量受到柱状图零点像素点数量的严重限制,水印容量没有得到充分挖掘。

本章介绍两种可逆数字水印算法:一种称为基于纠错编码的差值扩展可逆数字水印;另一种称为免疫数字水印。

### 6.5.2 基于纠错编码的差值扩展可逆数字水印

本算法属于强可逆数字水印算法。该算法在 Tian 算法差值扩展无损嵌入的基础上,保持了其高容量优点,同时纠正了过分修改像素对值、算法分类复杂、处理不对称等缺点。算法本着简化像素对分类,统一嵌入、提取和恢复过程,避免不必要的像素值修改的原则,在水印嵌入过程之前将嵌入对二值映射图 JBIG 压缩值和纯载荷一起进行纠错编码,将像素对仅分为可嵌入和不可嵌入两类处理。该算法是强可逆的无损数字水印算法。该算法能有效利用于医疗图像、遥感图像等数字产品保护。算法结构对称,实现简单,含印图像质量得到明显提高。

基于纠错编码的差值扩展可逆水印,是在 Tian 算法的基础上,采用差值扩展方法嵌入信息位。但改进之处在于大大简化分类原则,减少像素对不必要分类。本算法像素对仅分为两类:可嵌入类(I类)和不可嵌入类(II类)。在嵌入时,如果嵌入对是 I 类,则通过差值扩展嵌入数据;如果是 II 类,则不对像素对进行任何更改。这就大大减少了像素对分类处理的复杂度。对于 II 类像素对,由于嵌入时未对其进行任何修改,因此会造成提取时发生错误。因此,需要对 II 类像素对提取的数据进行纠错。纠错的前提是嵌入之前对隐体数据进行纠错控制编码。因此,本算法称之为基于纠错编码的差值扩展可逆水印。

隐体数据由恢复信息和纯载荷组成,其中恢复信息包括嵌入像素对在图像中位置的二值映射图 M 的 JBIG(二值图像压缩标准)压缩。在提取和载体恢复时,假定所有像素对均是 I 类,提取差值的最低位得到隐藏信息流。这种提取在 I 类像素对上不会发生错误,但在 II 类像素对可能会发生提取错误。因为 II 类像素对嵌入时未作任何更改,但提取时统一按 I 类处理,所以提取发生错误的概率为 0.5。这就需要对嵌入信息流进行纠错,得到纠错后

的信息流,从而首先恢复出恢复信息和嵌入载荷。恢复信息主要存储压缩的映射图  $M$ 。根据二值映射图  $M$ ,逐像素对地采用缩小差值方法对可嵌入像素对的像素值进行恢复;对不可嵌入像素对的像素值不作任何改动。

这里采用的差值扩展方法和 Tian 算法基本类似,下面举例说明原理。

### 1) 嵌入过程

假定有两个像素值  $x=206, y=201$ ,我们需要无损嵌入比特  $b=1$ 。首先要计算  $x$  和  $y$  的整型均值  $l$  和差值  $h$ :

$$l = \left\lfloor \frac{206 + 201}{2} \right\rfloor = \left\lfloor \frac{407}{2} \right\rfloor = 203$$

$$h = 206 - 201 = 5$$

其中, $\lfloor \cdot \rfloor$ 是向下取整函数。将  $h$  用二进制表示为  $h=5=(101)_2$ ,再将比特  $b$  追加到  $h$  的最不重要位(LSB)上,得到新值  $h'$ 。那么新的差值  $h'$  为  $h'=2\times h+b=2\times 5+1=11$ 。

用新的差值  $h'$  和原始的均值  $l$  来计算新值:

$$x' = 203 + \left\lfloor \frac{11 + 1}{2} \right\rfloor = 209$$

$$y' = 203 - \left\lfloor \frac{11}{2} \right\rfloor = 198$$

从嵌入对  $(x', y')$ ,可以提取出隐藏信息位  $b$ ,并恢复出原始的像素对  $(x, y)$ 。

### 2) 提取和恢复的方法

计算整型平均值和差值:

$$l' = \left\lfloor \frac{209 + 198}{2} \right\rfloor = \left\lfloor \frac{407}{2} \right\rfloor = 203$$

$$h' = 209 - 198 = 11$$

将  $h'$  采用二进制表示为  $h'=11=(1011)_2$ ,提取其 LSB,提取隐藏的信息位  $b$ ,然后得到原始的差值  $h=(101)_2=5$ 。公式表示为:

$$b = \text{LSB}(h) = 1$$

$$h = \left\lfloor \frac{h'}{2} \right\rfloor = 5$$

通过  $l'$  和恢复的  $h$ ,即可得到原始的像素对  $(x, y)$ 。

在上面的例子中,将差值  $h$  的长度由 3 位增加到 4 位以嵌入比特  $b$ 。该可逆嵌入过程  $h'=2h+b$ ,称为差值扩展(difference expansion,DE)。

下面介绍一下可扩展差值像素对和可更改差值像素对的概念。

为了通过差值扩展嵌入比特  $b$ ,新的扩展后的差值  $h'$  为

$$h' = 2h + b$$

为防止溢出,  $h'$  应该满足

$$|h'| \leqslant \min(2(255 - l), 2l + 1)$$

因此,对  $b=0$  或  $1$ ,有

$$2 \times h + b \leqslant \min(2(255 - l), 2l + 1)$$

这里将  $b=0$  或  $1$  都使上式成立的像素对称为可扩展差值像素对。

每个整数都可以表示为其最低位与去掉最低位后得到整数的 2 倍之和,即

$$h = 2 \times \left\lfloor \frac{h}{2} \right\rfloor + LSB(h)$$

其中,最低位  $LSB(h)=0$  或  $1$ 。如果更改其最低位为  $b'=0$  或  $1$ ,则

$$|h'| = \left| 2 \times \left\lfloor \frac{h}{2} \right\rfloor + b' \right|$$

为防止溢出,则对  $b'=0$  或  $1$ ,有

$$\left| 2 \times \left\lfloor \frac{h}{2} \right\rfloor + b' \right| \leq \min(2(255 - l), 2l + 1)$$

这里将对  $b'=0$  或  $1$  都使上式成立的像素对称为可更改差值像素对。

由定义易得出以下结论:

- (1) 对于可更改差值像素对,修改其  $LSB$  后,仍然为可更改差值像素对。
- (2) 所有可扩展差值像素对都是可更改差值像素对。
- (3) 差值扩展后的像素对是可更改差值对。
- (4) 当  $h=0$  或  $-1$  时,可更改差值和可扩展差值是等价的。

在 Tian 的算法中,像素对根据扩展能力分为四类:

- (1) EZ 类。包含所有可扩展差值的差值为  $h=0$  和  $h=-1$  的像素对。
- (2) EN 类。包含所有可扩展差值的但不属于 EZ 类的像素对。
- (3) CN 类。包含所有可更改差值的像素对,且不属于 EZ 和 EN 类。
- (4) NC 类。包含所有的不可更改差值的像素对。

在 Tian 的算法中,像素对总会归于上述四类中的一类。所有可更改差值像素对集合为  $EZ \cup EN \cup CN$ 。在构造已选扩展差值像素对位置映射图时,EZ 类嵌入数据;而 EN 类根据容量需要其中一部分用于嵌入数据,这一子集称为 EN1,另一部分称为 EN2。映射图尺寸为原图的一半。构造映射图时, $EZ \cup EN1$  赋予  $1$ , $EN2 \cup CN \cup NC$  赋予  $0$  值,即  $1$  表示实际数据嵌入点。

在嵌入数据过程中,Tian 算法需要嵌入以下数据:

- (1) 水印数据。
- (2) 压缩后的映射图。
- (3)  $EN2 \cup CN$  集合像素对的  $LSB$ 。

这些数据要嵌入到所有可更改像素对中:对  $EZ \cup EN1$  以差值扩展方式嵌入;对  $EN2 \cup CN$  以  $LSB$  替换嵌入。由此可见,CN 类像素对的嵌入不能增加净嵌入容量,但对其差值的修改使得含印载体质量增加了不必要的下降;而且像素对分类过于繁琐,增加了计算量。

为了避免 Tian 算法这种无效的过度差值更改,本文提出基于纠错编码的差值扩展无损嵌入方法。在本文算法中,像素对仅分为两类:可扩展差值像素对(记为 I 类),不可扩展差值像素对(记为 II 类)。满足像素对称为可扩展差值像素对,其余的像素对称为不可扩展差值像素对。映射图尺寸仍为原图的一半。构造映射图时,I 类赋予  $0$ ,II 类赋予  $1$ 。该映射图映射方法简单,压缩率高。

在嵌入数据过程中,本算法需要嵌入以下隐体数据:

- (1) 水印数据。
- (2) 压缩后的映射图。

嵌入前,我们要先对隐体数据进行纠错编码。实际嵌入时,对Ⅰ类像素对更改差值嵌入,对Ⅱ类像素对不做更改(相当于嵌入的数据有一半的错误概率)。这样,嵌入算法相当简单,而且避免了像素对值过度修改。

由上述算法描述可知,在嵌入时,Ⅱ类像素对实际未作更改。因此在提取时,Ⅰ类像素对数据提取完全正确;但对Ⅱ类像素对,数据提取错误概率大约为50%。因此,为了纠正提取错误(概率约为0.1%),需要对提取数据进行纠错。纠错的前提是对隐体数据进行纠错编码。

纠错码技术作为提高通信系统传输可靠性的有效手段,被广泛用于纠正数据经过不同信道传输后发生的错误。在图像数字水印系统中,可以利用这种错误保护技术来提高水印抵抗通信干扰的能力。常见的纠错码有汉明码、BCH码、Reed-Solomon码、LDPC码、卷积码、Turbo码等。其中汉明码、BCH码、Reed-Solomon码等分组码,主要适于离散无记忆信道,编译码性能较好和复杂度适中。

这里使用的纠错码是BCH码。BCH码是一种应用非常广泛的循环码,其纠错性能很好。BCH码还具有很好的代数结构,构造方便,人们根据所要求的纠错能力 $t$ 可以很容易地构造出BCH码。

对于任意整数 $m(m \geq 3)$ 和 $t(t < 2m - 1)$ ,存在 $GF(2)$ 上的BCH编码 $BCH(n, k, t)$ ,其中码长为 $n = 2m - 1$ , $k$ 为信息元的长度, $t$ 为纠错能力,即该码可以纠正 $n$ 个码元中的 $t$ 个或少于 $t$ 个错误。

为了有效地利用纠错码的纠错能力,尽量节省水印容量,需要根据Ⅱ类像素对在所有嵌入对中的分布确定相关参数,选择最优的BCH码。

假定嵌入数据流中比特用随机变量 $x$ 表示,嵌入数据流中0,1出现的概率是随机等概率的,即 $p(x=0)=p(x=1)=0.5$ 。而像素对差值的最低位中比特用随机变量 $y$ 表示,则差值最低位中0,1出现的概率也是随机等概率的,即 $p(y=0)=p(y=1)=0.5$ ,则Ⅱ类像素对中提取错误的概率为

$$\begin{aligned} p_{II} &= p(y=1 | x=0)p(x=0) \\ &\quad + p(y=0 | x=1)p(x=1) \end{aligned}$$

而这两种分布是无关的,因此

$$\begin{aligned} p_{II} &= p(x=0, y=1) + p(x=1, y=0) \\ &= p(x=0)p(y=1) + p(x=1)p(y=0) \\ &= 0.5 \times 0.5 + 0.5 \times 0.5 \\ &= 0.5 \end{aligned}$$

经过对USC-SIPI图像数据库[XXX]中3类305幅图像(Aerials、Miscellaneous、Textures)的处理,得到Ⅱ类像素对在总的像素对中的比例约为0.0023,所以总的错误提取概率

$$\begin{aligned} p_{\Sigma} &= \frac{N_{II}}{N_I + N_{II}} \times p_{II} \\ &= 0.0023 p_{II} \\ &= 0.0023 \times 0.5 < 0.0012 \end{aligned}$$

其中, $N_I, N_{II}$ 分别代表Ⅰ类和Ⅱ类像素对的数量。该错误提取概率相当于1000个提取位中大约只有1个错误提取。我们采用的是(255, 231, 3)的BCH码,其纠错能力足以保证能够

纠正所有的错误提取。

下面具体介绍水印嵌入与提取。

### 1) 水印嵌入过程

(1) 对原始载体图像生成像素对。

(2) 计算像素对差值

$$h = \{(h_{i,j}) \mid 1 \leq i \leq \lfloor m/2 \rfloor, 1 \leq j \leq n\}$$

(3) 计算二值映射  $M$  图。像素对属于 I 类, 则为 0; 像素对属于 II 类, 则为 1。

(4) 对  $M$  图进行 JBIG 压缩, 得到  $M'$ 。

(5) 将  $M'$  图作为头部, 水印信息  $W$  为主体, 生成隐藏信息  $W'$ 。

(6) 对隐藏信息进行纠错编码, 得到编码后隐藏信息  $W''$ 。

(7) 对  $W''$  进行置乱, 得到隐藏信息  $W'''$ 。

(8) 采用扩大差值方式将  $W'''$  嵌入。

(9) 重新计算扩差后的像素对值, 得到水印后图像。

### 2) 水印提取过程

(1) 生成水印后图像的像素对。

(2) 计算像素对差值和均值。

(3) 提取像素对差值二进制表示最低位, 得到提取序列。

(4) 对提取序列进行纠错解码, 得到解码数据。

(5) 解码数据头部为二值映射图, 解码数据主体为水印信息。

## 6.6 免疫数字水印算法

免疫数字水印算法, 是一种新型的弱可逆数字水印。根据水印保护作品的特点, 有些情况下并不苛求可逆数字水印将含印作品完全无差别地恢复到原始状态, 而是允许一定的微小差异存在。在此前提下本文介绍一种数字水印实现的新思路, 要求水印嵌入后作品质量下降到失去使用价值: 这样公开的嵌入水印的作品对攻击者没有利用价值; 如果用户得到授权, 就可以从公开作品中自恢复得到高清晰度作品。该水印技术因为公开作品的不可利用性, 使各种攻击(如移除攻击、覆盖攻击、变换攻击等等)失去了攻击价值; 即使遭到攻击, 因为攻击破坏了作品的恢复码, 将不可能得到高保真作品。因此该数字水印有天生的免疫性, 能有效遏制非法复制和篡改, 能够直接有效地保护数字产品。区别于传统水印, 本文称之为自恢复载体图像的免疫数字水印(self-recovery image immune digital watermarking, SRIW)。版权所有者公开自己的免疫性含水印作品, 使用户可以通过公开作品对其进行初步了解; 用户只有在得到授权证书的情况下, 才能从公开作品恢复出高保真作品。这种技术能很好地应用到数字产品的版权保护中。

### 6.6.1 SRIW 形式化描述

SRIW 中公开的图像是没有使用价值的图像, 但其包含自恢复码, 称为公开免疫图像。授权用户可以从中恢复出清晰的高保真度图像, 恢复出的高保真图像称为自恢复图像。SRIW 应该满足以下条件:

$$\begin{aligned}
 \text{Img}_{\text{pub}} &= E_1(\text{Img}_{\text{ini}}, W, \text{Key}) \\
 \text{Img}_{\text{rec}} &= f(\text{Img}_{\text{pub}}, \text{Key}') \\
 \|\text{Img}_{\text{rec}} - \text{Img}_{\text{ini}}\| &\rightarrow O \\
 \|\text{Img}_{\text{pub}} - \text{Img}_{\text{ini}}\| &\gg \|\text{Img}_{\text{rec}} - \text{Img}_{\text{ini}}\| \\
 \|\text{Img}_{\text{pub}} - \text{Img}_{\text{ini}}\| &> T
 \end{aligned}$$

其中,  $\text{Img}_{\text{ini}}$  是原始载体图像;  $\text{Img}_{\text{pub}}$  是公开的含水印免疫图像;  $\text{Img}_{\text{rec}}$  是自恢复后的图像;  $W$  是版权标识水印;  $E_1$  是免疫嵌入函数;  $\text{Key}$  是水印嵌入密钥;  $\text{Key}'$  是免疫图像恢复密钥;  $\|\cdot\|$  是任意矩阵范数;  $T$  是免疫门限。

## 6.6.2 SRIW 实现方法

SRIW 的优点在于其免疫性和自恢复性。其关键在于以较高嵌入深度植入破坏性水印, 并生成补偿矩阵, 同时将补偿矩阵本身作为二次水印植入图像中。SRIW 的实现不依赖于实现方法, 本文以小波域为例简述其实现过程。

水印生成和嵌入过程如下: 图像的数据可以区分为重要信息域(important area, IA)和非重要信息域(unimportant area, UA)。将图像  $I$  进行一级 Haar 小波变换, 图像的能量集中在逼近子图 LL, 为重要信息域 IA; 子带 HH 为图像高频细节信息, 为非重要信息域 UA。将水印( $K_1$  加密)嵌入到图像重要信息域 IA, 保证水印的鲁棒性; 同时计算 IA 的自恢复码(SRC)。将 SRC 用  $K_2$  加密得到 SSRC, 嵌入到 UA 区域。然后进行逆小波变换, 得到公开的含水印免疫图像(public immune image)用  $I_p$  表示。公式为:

```

W=GEN(M)           //生成水印
[IA,UIA]=DWT(I)   //对图像进行小波变换,得到 UA 和 IA
IA'=IA(1+αW)       //将水印以高嵌入比 α 加性植入到 IA 区域
SRC=IA-IA'          //生成 IA 的补偿矩阵
SSRC=C(SRC,K)      //补偿矩阵加密
UA=G(SSRC,UA')      //将补偿码嵌入不重要区域
Ip=IDWT(IA,UA')    //逆小波变换,得到免疫图像
  
```

图像恢复和水印提取过程如下: 用户取得公开的  $I_p$  图像后, 可以进行初步内容查看。如果需要, 向作品提供者申请, 获得授权证书(含加密的用户信息和图像恢复密钥  $K'$ )。对图像  $I_p$  逆小波变换, 从  $UA'$  解析出 SSRC。利用 SSRC 和恢复密钥  $K'$  恢复出图像 IAR 区域, 将图像进行逆小波变换, 得到图像的高保真度图像 IR。水印是通过 IA/IAR-1 盲提取得的。公式表示为:

```

[IA',UA']=DWT(Ip)           //对模糊图像进行小波变换
[SSRC,UAR]=IG(UA')         //从 UA' 中解析出 SSRC 和 UAR
SRC=IC(SSRC,K')              //解密得到自恢复码 SRC
IAR=IA'+SRC                //补偿得到重要区域编码
IR=IDWT(IAR,UAR)        //逆小波变换得到高保真度恢复图像 IR
W=IA'/IAR-1                //得到水印
M=IGEN(W)                    //得到版权标识消息
  
```

需要指出的是, SRIW 的嵌入深度比传统的水印要高, 可选择的范围也要大得多。只要根据不同的需要, 选择相应的嵌入深度, 就能得到合适的免疫水印。

### 6.6.3 SRIW 安全性分析及评价标准

SRIW 的作品质量不清晰, 可能会引起攻击者的注意或兴趣。由于 SRIW 公开的免疫图像没有利用价值, 常见的攻击形式(如移除攻击、覆盖攻击、几何变换等)都失去作用。真正有威胁的攻击形式是破解自恢复码。因此, 这种算法要求对水印信号和自恢复码 SRC(补偿矩阵)的安全性要求很高。SRIW 必须满足两项安全条件:

- (1) 在假定算法已知的情况下, 攻击者无法得到补偿值水印, 从而无法得到清晰作品。
- (2) 在假定算法已知的情况下, 攻击者无法得到或移出版权认证水印。

条件(1)是算法安全性的核心所在, 是条件(2)的基础和保证。条件(1)、(2)的必要前提是版权认证水印和自恢复码 SRC 的预处理和加密算法是高安全强度的。实际上, 在算法公开的情况下, SRIW 的安全性等价于 SRC 的加密算法。因此, 要保证 SRIW 的安全性, 需采用可信任的高强度加密算法, 对 SRC 进行加密, SRIW 可以达到等同的安全强度。算法的密钥必须严格保密, 预处理算法本身最好也是保密的。保证攻击者在正常情况下, 无法得到自恢复码 SRC, 从而无法恢复出清晰作品。

本文所采用的方法是在补偿值嵌入之前, 对补偿矩阵 SRC 进行 N 次(作为密钥的一部分)Arnold 置乱, 得到置乱矩阵。再将置乱矩阵序列化后, 对其进行高级加密标准 AES 算法(密钥 KS), 进行加密处理。之后进行反序列化, 得到加密矩阵 MSRC, 真正嵌入 UA 的自恢复码是 MSRC。对 N 和密钥 KS 进行 1024 位 RSA 算法进行高强度加密, 作为授权证书的一部分。自恢复码 SRC 提取时, 是上述处理的逆过程。用户只有得到授权证书, 才能解密自恢复矩阵, 得到清晰图像。

由于 SRIW 的自身特点, 传统的水印评价(如隐蔽性、鲁棒性和脆弱性等)标准不能完全适用于这种新型的水印技术, 需要 SRIW 特有的评价标准。SRIW 特有的评价指标主要包括:

- (1) 公开免疫作品质量(quality of public immune works)。记为  $Q_p$ , 可用公开免疫作品图像与原始图像的峰值信噪比  $PSNR_p$  表示。
- (2) 可适用恢复质量(quality of applicable recovery)。保证公开作品免疫的情况下, 作品能够自恢复的质量, 记为  $Q_R$ 。可用自恢复图像与原始图像的峰值信噪比  $PSNR_R$  表示。
- (3) 最优可适用恢复质量(max quality of applicable recovery)。保证公开作品免疫的情况下, 作品能够自恢复的最高质量, 记为  $\lceil Q_R \rceil$ 。
- (4) 最差可适用恢复质量(min quality of applicable recovery)。保证公开作品免疫的情况下, 作品能够自恢复的最低质量, 记为  $\lfloor Q_R \rfloor$ 。
- (5) 恢复质量差值(diff quality of applicable recovery)。自恢复作品与免疫作品之间的质量差值, 表明免疫作品的自恢复能力, 记为  $|Q|$ 。显然  $|Q| = Q_R - Q_p$ 。
- (6) 嵌入柔性(embedding flexibility)。保证公开作品免疫的情况下, 水印嵌入深度的范围, 即最大可适用嵌入与最低可嵌入深度的差值, 记为  $F_a$ 。

为了验证免疫水印, 本文在 Matlab 7.1 下进行了仿真实验。使用的原始载体图像和

水印图像如图 6.10 所示。图 6.11~图 6.13 三组图像分别是三次不同嵌入深度的实验结果。

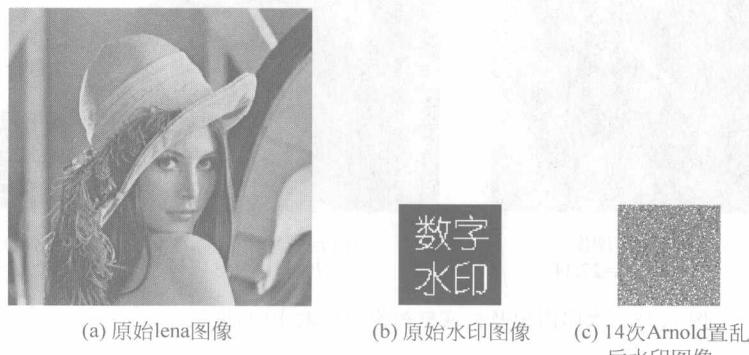


图 6.10 原始载体和水印准备



图 6.11 水印图像及自恢复图像和提取水印, 嵌入深度  $\alpha=0.015$



图 6.12 水印图像及自恢复图像和提取水印, 嵌入深度  $\alpha=0.15$

从三组不同嵌入深度的结果图像可以看出：

- (1) 嵌入深度小于 0.05 时, 含水印图像很清晰, 属于传统水印。
- (2) 嵌入深度大于 0.3 时, 恢复图像效果不太好, 用途不大。



图 6.13 水印图像及自恢复图像和提取水印, 嵌入深度  $\alpha=0.40$

(3) 嵌入深度在 0.05 与 0.3 之间时, 含水印图像清晰度差, 失去欣赏价值, 但能了解图像基本内容; 经授权后从其自恢复的图像清晰度非常高。这种水印就是自恢复免疫水印。

采用 SRIW 自恢复免疫水印技术, 可以将含水印模糊作品免费公开, 帮助潜在用户了解作品大致内容, 用于作品自身宣传和传播。授权用户可以通过公开的模糊图像获得高保真度的自恢复作品。SRIW 的优势在于公开免疫作品能天然抵抗常见的水印攻击。因为常见的攻击只会破坏脆弱的自恢复码, 从而使攻击者无法得到有价值的高清作品, 有天然免疫性。

SRIW 嵌入深度选择柔性很大。在大多数文献中, 传统水印因为高保真度要求, 嵌入比上限很低, 0-1 二值水印嵌入一般不超过 0.05, 造成嵌入的范围选择很小, 同时降低了传统水印的鲁棒性和安全性。

## 6.7 多重数字水印

### 6.7.1 多重数字水印概述

目前大多数水印算法都支持单水印嵌入, 但单水印在实际应用中有一定的局限性。例如鲁棒性水印可以有效地进行版权认证, 但无法证明对作品的篡改; 半脆弱性水印则反之, 可以有效地进行作品内容的完整性认证, 但对水印攻击的抵抗能力很弱。一个很自然的思路就是, 如果能结合不同类型水印的优点, 在作品中嵌入多种类型水印, 可能会更有效地进行版权保护。

还有一种情形是在多人共享版权时, 需要同时嵌入多个用户的不同水印。这些水印要求具有同等的版权认证能力, 同时支持独立提取和联合认证。但目前还没有能很好满足以上需求的多用户水印算法。

针对以上需求, 多重水印逐渐得到重视。最初的多重水印算法大都是对单水印算法进行改进, 使之能够支持多重水印算法。

Wong 提出了三种多重盲水印算法, 即单水印多密钥 SWE 算法、多水印共享水印空间算法 MWE 和基于 JPEG 的迭代多水印算法 IWE。Liu 将多重水印看做广播的脏纸通信, 同时指出水印的能量共享效率要高于分时共享。Giakoumaki 将访问控制、源标识等 6 种信息同时嵌入医学图像的不同层次的小波系数中。Takahashi 提出了采用相位调制的音频信

号多水印算法,利用人类听觉系统(HAS)对声音相位微调的不敏感性,同时嵌入“版权管理信息”、“版权控制信息”、“数字指纹”三种水印信息。这三种水印采用 FDM 进行调制,然后一起嵌入到载体中。

综合目前大多数多重数字水印算法,其实现方法可以归为以下几类:

- (1) 多水印组合后嵌入。
- (2) 多水印共享信道同时嵌入。
- (3) 多水印采用各自独立信道嵌入。
- (4) 多水印重叠嵌入。

目前的多重水印算法多归为第一类,实际上可以看做是单水印嵌入,不支持用户独立认证。

大多数多重水印算法存在水印碰撞、多重水印叠加嵌入的相互影响、用户数量增加时含印作品质量严重下降从而限制了用户数量等问题。主要根据多用户共享版权时独立认证,鲁棒性、完整性要求,分析基于 CDMA(code division multiple access,码分多址访问)的共享信道多用户水印算法,能有效解决多用户共享版权时的作品版权共享认证和多种水印联合嵌入。虽然目前也有不少 CDMA 数字水印的文献,但大都是将 CDMA 用于水印的扩频来增强鲁棒性的单水印算法,并没有充分发掘 CDMA 支持多用户共享信道的优良特性。

### 6.7.2 鲁棒性和脆弱性相结合的双重数字水印

本算法是基于小波变换的双重并行嵌入水印算法,算法利用鲁棒性水印进行版权认证,利用脆弱性水印进行完整性认证。鲁棒性水印所关心的是自身的完整性,而半脆弱性水印鉴定的是载体作品的内容完整性,能够证明对作品的篡改。

双重水印的嵌入是在小波域进行的,对图像进行小波全局变换后,同时嵌入鲁棒性水印和脆弱性水印,达到版权保护和完整性认证的目的,其中的脆弱性水印能够对篡改进行定位。水印的提取不需要原图像。该技术为数字作品提供了完整且安全的保护方案。仿真实验结果表明,该方法同时具有很好的稳健性、安全性和对篡改的识别能力。

首先对图像进行小波变换,在图像的低频嵌入稳健水印(标识),在高频嵌入脆弱性水印(因为小波变换是全局变换,而且高频反映的是细节部分,所以对图像的任何修改都会引起脆弱性水印的破坏)。但由于高频率带系数过小,直接嵌入脆弱水印会造成脆弱水印未受攻击就提取失败。解决方案是对低频子图( $d_1^1$ )进行二次小波变换,再次得到  $d_1^1$  的四个子图,即  $d_2^1, d_2^2, d_2^3, d_2^4$ 。这些子图对离散性带来的误差更具有鲁棒性,所以我们将鲁棒性水印嵌入  $d_2^1$ ,将脆弱性水印嵌入到  $d_2^4$ 。两级小波变换如图 6.14 所示。

为了增强水印的安全性,水印的嵌入和提取均需使用密钥。密钥由两部分组成:①前半部分保存鲁棒性水印的置乱参数和嵌入控制参数。先对水印图像进行置乱(Arnold 变换),同时生成混沌序列,利用混沌序列控制水印的嵌入深度。②密钥后半部分保存脆弱性水印控制参数——混沌的初始值和参数。利用初始值和参数生成混沌序列。利用该序列控制水印比特序列的嵌入深度。

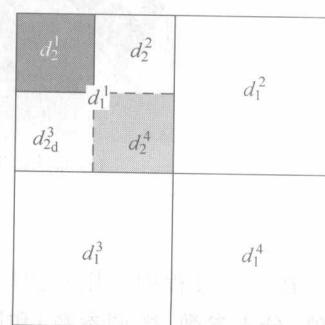


图 6.14 两级小波变换

利用密钥生成鲁棒性水印和脆弱性水印的混沌序列，并利用混沌序列生成嵌入深度的控制矩阵。之后用控制矩阵控制两个水印的嵌入深度。鲁棒性水印和脆弱性水印嵌入方法相同，均采用量化嵌入方式。

嵌入的具体步骤为：

(1) 对原始载体图像进行两级小波变换。其中第二级小波变换是在第一级小波变换的低频子图  $d_1^1$  上进行的。

(2) 对鲁棒性水印和脆弱性水印进行 Arnold 变换，变换次数  $\eta_1$  和  $\eta_2$  作为密钥。

(3) 将鲁棒性水印  $w_1$  嵌入到  $d_1^1$  子图系数中。利用混沌序列  $L_1$  生成控制矩阵  $M_1$ 。采用  $M_1$  控制水印序列  $w_1$  的量化参数。

(4) 将脆弱性水印嵌入到  $d_2^4$  子图系数中。采用混沌序列 X 控制水印序列  $w_2$  的量化参数。

提取的具体步骤如下：

采用密钥生成鲁棒性水印的控制矩阵和脆弱性水印的混沌控制序列。利用控制矩阵和混沌序列分别提取两个水印。

(1) 对含印载体图像进行两级小波变换。第二级小波变换是在第一级小波变换的低频子图  $d_1^1$  上进行的。

(2) 利用混沌序列生成控制矩阵  $M_1$ 。利用  $M_1$  控制提取时的量化参数，计算出鲁棒性水印  $w_1$ 。

(3) 对提取的水印进行 Arnold 逆变换，逆变换的次数为  $\eta_1$ ，从而得到最终的鲁棒性水印。

(4) 从  $d_2^4$  子图系数中提取脆弱水印。提取时采用混沌控制序列 X 控制量化参数。

(5) 对提取的水印  $w_2$  进行 Arnold 逆变换，逆变换的次数为  $\eta_2$ ，从而得到脆弱性水印。

为直观地验证双重数字水印的效果，我们采用在 Matlab7.1 上实现上述算法。实验采用的载体图像是“Airplane”，如图 6.15(a)所示；嵌入的鲁棒性水印和脆弱性水印如图 6.15(b)和图 6.15(c)所示。



图 6.15 原始载体和原始水印图像

在嵌入过程中采用密钥增强安全性。密钥由水印图像置乱次数、混沌控制序列参数(初始值、分支参数、微调参数)和嵌入深度值组成。嵌入鲁棒性水印和脆弱性水印时采用不同的密钥。这样，在水印提取时，必须使用正确的密钥才能正确提取。含印载体在嵌入双重水

印后,保持了较高的质量( $PSNR$ 为54.33)。鲁棒性水印提取达到0误差;而脆弱性水印因为水印处理过程中的量化作用而有轻微误差存在( $MSE$ 为0.012)。如果没有密钥或者密钥错误,那么攻击者就无法得到嵌入的水印。这说明,算法安全性是基于密钥的。

本算法中嵌入了两个水印,二者性质和应用是互补的:我们可以在含印载体遭到内容篡改的情况下,仍然能够提取出鲁棒性水印,这样可以用来证明作品的版权;而脆弱性水印在受到攻击后即提取失败,这可以用来提示含印载体受到了非法篡改,从而对作品进行完整性验证。二者结合起来,可以同时提供作品版权认证和完整性验证的较完整的应用方案。

图6.16是上述算法的正确提取的鲁棒性水印和脆弱性水印。图6.17是上述算法中如果密钥错时,提取的鲁棒性水印和脆弱性水印。图6.18是上述算法遭裁剪攻击时提取的鲁棒性水印和脆弱性水印。图6.19是遭篡改攻击时上述算法提取的鲁棒性水印和脆弱性水印。



(a) 含印载体图像

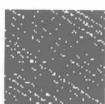


(b) 提取的鲁棒性水印



(c) 提取的脆弱性水印

图6.16 含印载体和正确提取的水印图像

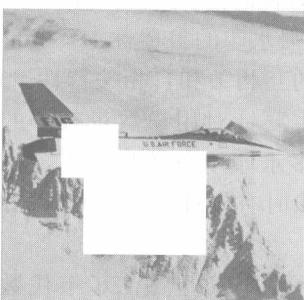


(a) 错误提取的鲁棒性水印



(b) 错误提取的脆弱性水印

图6.17 密钥错误时提取的水印



(a) 遭裁剪攻击含印载体图像



(b) 提取的鲁棒性水印



(c) 提取的脆弱性水印

图6.18 遭裁剪攻击时提取的水印

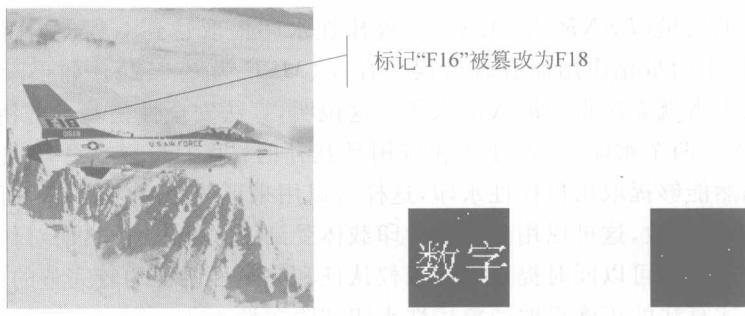


图 6.19 遭篡改攻击时提取的水印

### 6.7.3 基于 CDMA 的多重数字水印算法

QED 大多数多重水印算法存在水印碰撞、多重水印叠加嵌入的相互影响、用户数量增加时含印作品质量严重下降从而严重限制了用户数量等问题。根据多用户共享版权时独立认证、鲁棒性、完整性要求,本文提出基于 CDMA 的共享信道多用户多重数字水印算法。该算法能有效解决多用户共享版权时的作品版权共享认证和多种水印联合嵌入。虽然目前也有不少 CDMA 数字水印的文献,但大都是将 CDMA 用于水印的扩频来增强鲁棒性的单水印算法,并没有充分发掘 CDMA 支持多用户共享信道的优良特性。

CDMA 通信是将相互正交(或准正交)的不同编码分配给不同用户来调制信号,通过码分多址实现多用户同时通过共享信道进行通信。CDMA 在无线通信中得到了广泛的应用,它具有可多址复用、容量大、保密性好、抗干扰能力强和抗噪声等优点。

基于 CDMA 在共享信道中为不同用户分配正交码来实现多用户同时访问的思想,分析为共享作品版权的多个用户分配各自正交码,对不同用户的水印信号进行 CDMA 编码,从而实现多用户共享作品版权的多重数字水印。该算法具有以下突出优点:

- (1) 能有效解决多重数字水印中的水印碰撞和相互影响问题。
- (2) 用户数量增长不会造成含印作品质量严重下降,解决用户数量受容量所限问题。
- (3) 由于利用相互正交(或尽可能正交)的编码去调制信号,会将原信号的信号频谱带宽扩展,因此,对这种调制方式具有扩频作用,同时能增强水印的鲁棒性。

在扩展频谱的 CDMA 系统中,要求扩频码有良好的互相关和自相关特性。像 m-序列、Gold 码、LS 码等具有很好的自相关和互相关特性,但主要针对用户数量较多的 CDMA 无线通信,生成正交码计算复杂度大。就水印应用而言,多水印用户数量有限,主要保证用户水印之间的正交性,扩频要求不是主要目的,所以在同步时有较理想的互相关特性的 Walsh 正交码足以满足应用需求。

Walsh 函数(或序列),通信中也称为 Walsh 码。是由 20 世纪 20 年代数学家 J. L. Walsh 提出的。Walsh 函数是定义在归一化区间(0,1)的完备正交函数集,除了在有限个不连续点上取 0 值外,每个函数取值仅为 +1 或 -1。按 Walsh 顺序排列的 Walsh 函数定义如下:

$$Wal(K, t) = \prod_{r=0}^{p-1} \text{sgn}(\cos k_r 2^r \pi t), t \in [0, 1], K = \sum_{r=0}^{p-1} k_r 2^r$$

上述公式中  $k_r$  是序数  $k$  的二进制表示中各位二进制数字的值。利用公式可以很容易求出各个 Walsh 函数。工程应用中更常用的一种方式是用 Hadamard 变换产生 Walsh 码：

$$H_1 = [1]$$

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

由上面生成的 Hadamard 矩阵,其行向量或列向量之间都是正交的 Walsh 码。

在实际应用时应根据用户数量,设定 Hadamard 变换的次数  $N$ ,生成相应数量和长度的 Walsh 码,然后为每一用户分配一个 Walsh 码  $C_k$ 。CDMA 水印编码时,为了平衡混合信号,将原始水印信号由 {0,1} 映射为 {-1,1}。即水印序列的点值为 1 时,编码为用户正交码本身;水印序列的点值为 0 时,编码为用户正交码的反码;若编码为 0,表示用户未发出水印信号。

在水印嵌入时,用户先用各自的 Walsh 码  $C_k$  对自己的水印信号  $M_k$  进行编码,然后所有编码后的水印信号  $C_k M_k$  进行多路混合,得到多路混合水印信号  $W$ ,再进行水印嵌入。水印生成和嵌入过程的公式为

$$W = \sum_{k=1}^N C_k M_k$$

$$X' = X + \alpha W$$

式中,  $N$  为用户数量;  $C_k$  是分配给第  $k$  个用户的正交码;  $M_k$  是第  $k$  个用户的水印。

在水印提取时,用户根据自己分配的正交码  $C_k$ ,可独立提取各自的水印。用户采用各自的正交码进行相关计算,将自己的水印信号从多路混合水印信号中分离出来。水印提取和分离的公式为

$$\widetilde{W} = (X' - X)/\alpha$$

$$\widetilde{M}_k = (\widetilde{W} \otimes C_k) / (C_k \otimes C_k)$$

式中,  $\widetilde{M}_k$  即为提取出的第  $k$  个用户的水印。

## 6.8 本章小结

本章主要研究了典型的数字水印算法,着重介绍了可见与不可见数字水印算法、可逆数字水印算法、免疫数字水印算法和多重数字水印算法。

## 6.9 复习题

- 6.1 常用的数字水印算法如何分类?
- 6.2 空域与变换域存在着何种不同?各自的优缺点是什么?
- 6.3 如何实现 DCT 变换?
- 6.4 自己设计一种可见水印的代码,并在相应环境中实现。
- 6.5 完成可逆水印的嵌入和提取代码。

## 隐蔽通信

### 本章目标

- 理解隐蔽通信技术。
- 了解隐蔽信道。
- 理解 TCP/IP 的隐蔽通信。
- 理解其他协议如何实现隐蔽通信。

### 7.1 隐蔽通信概述

随着 Internet 的广泛使用,人们对安全的要求越来越高,安全已经成为每个人的需求,直接或间接地同网络环境相关。除了软件方面的解决方案之外,信息隐藏、密码术和网络安全的结合为隐蔽通信提供了具体的环境,目前这是一个活跃的研究领域。信息隐藏,除了和 Internet 上的数字媒体相关,很显然还与通信息息息相关。所以必须将传统的安全解决方案与网络结合起来,是网络与信息隐藏的结合,是信息隐藏研究的一个重要领域——隐蔽通信。

在信息隐藏的预处理,经过嵌入算法形成伪装载体之后,伪装载体需要通过开放的环境进行直接或间接的传输,如果为了实现安全模型中最后一层的安全,就必须使用隐蔽通信,在所有研究信息隐藏的文献中,都必不可少地提到隐藏通信,但有关隐蔽通信的论文很少,文献少的原因也可能出于保密的原因没有公开。隐藏通信可以引用的例子包括国下信道、隐蔽通道等。国下信道的概念首先是由数学博士 Simmons G. J. 于 1978 年在美国圣地亚国家实验室(Sandia National Labs)提出的,当时国下信道提出的目的在于证明美国用于第二阶段限制战略武器谈判条约 SALT-II 核查系统中的安全协议的基本缺陷,Simmons 给出了一个描述性的定义:国下信道存在于诸如密码系统、认证系统和数字签名方案等加密协议中,该信道在发送者和隐藏的接收者之间传送秘密的信息,该信息不能被公众和信道管理者所发现。另外,在网络通信中,去跟踪敌手的数据包,进行通信量分析,以及判断通信双方的身份,是收集谍报信息的一个重要来源,而采用隐蔽通信的技术的主要目标就是保护通信信道不被别人窃听和进行通信量分析,这种技术提供了一种基于 TCP/IP 协议的匿名连接,从数据流中除去用户的标识信息,用该技术建立连接时,并不是直接连到目的主机相应的数据库,而是通过多层代理服务器,层层传递后到达目的地址,每层路由器只能识别最临近的一

层路由器,第一层路由器对本次连接进行多层加密,以后每经过一层路由器,除去一层加密,最后到达的是明文,这样每层路由器处理的数据都不同,使敌手无法跟踪,连接终止后,各层路由器清除信息。这种技术可用于有线电话网、卫星电话网等。可广泛使用于 E-mail、Web 浏览以及远程注册等。研究隐蔽通信中实现信息隐蔽的葱头路由器(Onion Router);研究开发路由器接收、处理数据所采用的多层加密及分层解密技术;研究如何从数据包中分解通信的参与者的身份及地址信息,并隐匿通信的通信量。军事级别的依赖于公共的通信基础设施,所以在网络通信领域有着不可估量的应用前景。

这里所研究的隐蔽通信是指在公开的信道中所建立的一种实现隐蔽通信的信道。利用系统接入控制机制中的漏洞建立起来,并实现隐蔽通信的技术,隐存储信道、隐定时信道和阈下信道。前两种并称为隐蔽信道,通过采用特殊的编译码方式使不合法的信息流逃避常规的安全控制机构的检测来实现。隐信道可以采用不同的文件名,或多个身份等信息来编码秘密消息,隐蔽性很强。隐定时信道可以利用时间轴上的事件序列来进行编码,隐信道的存在条件大致如下:网络设计实现中的漏洞,如文件命名的规定,用户编码等接入控制机构实现或运行的不正确所造成的漏洞;接收双方之间存在共享资源。系统中被植入特洛伊木马等。只要在网络上接收者和发送者之间存在共享的资源就都可以建立隐蔽信道。

在网络通信中,信息隐藏很自然地映射成一个通信问题,发信者成为密信信源,收信者成为密信信宿,密信经过的通道构成了隐蔽通信信道。隐蔽通信不仅使通信成为秘密,并且不可见。

## 7.2 隐 蔽 通 道

隐蔽通道的概念首先由 Lampson 提出。Lampson 确定地描述了隐蔽通道可以用于信息传输,但是他没有具体实现,这个理论也没有具体用于通信。之后有许多对这个基本概念的进一步分析。这些分析详细描述了相关的隐蔽通信的概念,相关的资源分配的策略,在不同系统安全级别的共享资源,资源变化状态和资源管理实现。

这些方面都与系统内发生的通信相连接。在系统中可用资源的状态和系统可用的资源都能用于隐蔽通道:即在系统内部从一方到另外一方发送信号信息。变化表明文件状态在系统中有一系列的点状态。在更完整的定义中提供了包括隐蔽通道的访问控制策略的可能性和它的实现。隐蔽通道是描述双方之间的通信连接,它允许一方传送信息到另一方,以不违反系统的安全策略方式。隐蔽通道分为两类:隐蔽存储通道和隐蔽时分通道。在隐蔽存储通道中的通信发送方必须把隐藏数据写入到存储区(不是通信的存储区),并且信息的恢复由接收方完成。相反,在隐蔽时分通道中,通信需要发送方通过调制自身系统资源发送信号信息。隐蔽通道可以利用源方和接收方的嵌入和检测处理的结合。通过定义,隐蔽通道的存在一定是不可检测的。

隐蔽通道是信息隐藏的一个主要分支。在信息隐藏中,通信双方允许彼此通信,基于系统的安全策略,当使用隐蔽通道定义相关的特性时,即在合法的信息内容上加上无法察觉的信息。这就导致了隐写术学科的出现,隐写术起源于希腊的隐写,隐写术就是隐蔽信息的存在,将秘密信息藏入无知的伪装信息中。最简单的例子通常是指在数字图像中使用每个像素的低端两个或三个比特位来隐蔽信息并进行通信。因为最后的两或三位的信息不影响载

体图像内容，并且隐藏了秘密内容的存在。因此这同样适用于从接收方到发送方的隐蔽通信。因此隐写术确保了隐蔽通道用于传送秘密信息。

从网络通信的观点，隐蔽通信也能使用数据包作为载体。因为所有通过网络的信息都以数据包的形式通过网络，当通过不同网络拓扑，在它们到达目的地之前，这些数据包由网络结点共享。在网络环境中信息隐藏的完善的方案应该围绕网络并结合信息隐藏。

隐蔽通道的定义违背了系统安全的策略。因此这样的通道威胁到系统安全。另一方面没有被利用的带宽的可用性使这些隐蔽通道存在。本章的目的是研究隐蔽通道，就是要研究这些没有被利用的带宽以及相关的正在使用的各种网络程序和机制。

## 7.3 TCP 隐蔽通信

TCP/IP 协议为通信提供了语法和语义规则。它们包括信息格式的详细资料，描述了当信息到达时计算机如何响应，特别说明计算机如何处理差错或其他异常条件。更重要的是，它们推动了计算机通信独立于双方网络的硬件。协议对通信而言，就是计算方法。从上面的分析中我们知道，隐蔽通信与 TCP/IP 组以及网络协议相关，如 IGMP、ICMP 等，对于这些协议中的每个协议，都可以实现隐蔽通信。深入的隐蔽通道分析重要的是对 IP 协议和与它相关的安全机制的分析，另外还涉及 IP 包处理以及伪装数据包的排序处理。

### 7.3.1 TCP 协议概述

TCP/IP 协议组能够提供简单开放的通信基础设施。目标是通信的最大化、连通性和协作。组是分等级的协议，它由交互模块组成，每个交互模块提供具体的功能。它是基于方便的包交换技术，但是不依赖于任何特定厂家的硬件。这组协议的重要意义在于它从网络技术中独立出来和它的通用互连，只要计算机的双方都使用 TCP/IP 协议，双方就能通信。

协议组提供了三种服务，面向应用的服务、可靠的服务和无连接的服务。可靠的和无连接的信息传送由网络层服务提供。而面向应用的机制由应用层提供。后者的服务提供了一系列应用程序，它使用底层网络来携带有用的通信任务。最流行的 Internet 应用服务包括：WWW、E-mail、文件传输和远程登录等。

无连接服务包括网络数据包的尽力而为的传送，它是最基本的互联网服务，TCP/IP 根据信息携带的地址信息，使用网络从一台计算机向其他计算机发送信息，这里包传送并不承诺必须到达目的地。

IP 协议位于网络层，它提供无连接服务。

可靠的传输层服务允许在一台计算机上的应用程序建立连接，同另外不同的计算机来传送大量数据并通过表面上相联的硬件连接。因此，可靠的传输层服务确保包发送到目的地，并且没有传输差错、丢失包和中间结点的失败，确保传输沿着从发送方到接收方的路径。TCP 提供了这些可靠的传送服务并形成了传输层协议组。网络分层结构如表 7.1 所示。

表 7.1 网络分层结构

应用层	FTP, Telnet, DNS, SMTP
传输层	TCP, UDP
网络层	IP, ICMP, IGMP
数据链路层	网络接口和设备驱动

在最高层是应用层,它为相关程序访问网络层服务提供了窗口。下面两层执行协议栈的主要功能。传输层负责在两端点间可靠透明地传送数据。TCP 和 UDP 位于传输层。网络层主要提供地址和字段的传送工作,使数据能通过网络到达目的地。IP、ICMP 和 IGMP 位于协议栈的网络层。数据链路层再现实际网络硬件的通信(如网卡)。这就是不同的驱动对应不同的接口,我们关心的是隐蔽。

在开发 TCP/IP 期间,很少注意安全方面。例如 TCP/IP 并没有确保传送信息的完整性。也没有对传送包的来源验证。TCP/IP 正式的模型面临的安全威胁出现了。如 IP 口欺骗、TCP 会话劫持,死亡之 Ping, TCP 序列号预测和 SYN 泛洪攻击等。这些模型表现了 TCP/IP 安全技术,并能更好地理解内在的弱点。同样指出了一系列安全缺陷在 TCP/IP 协议组中,并详细地描述了一系列的进攻。另外确定了威胁,也介绍了许多防御,如加密等。

在一些具体的例子里,在协议中引入冗余。部分的规范有助于对安全弱点进行保护。另外,这有多种 TCP/IP 设计策略的解释,本质上使用冗余。但是对隐蔽通信来说,冗余是通信的关键因素,不直接面临安全威胁,TCP/IP 协议组也易受标准执行的隐蔽通信的影响。

我们的工作重点是在 TCP/IP 环境中隐蔽通道存在的问题,通过介绍一系列信息隐藏情况。隐蔽通道被认为是对系统安全的潜在的威胁。但是,我们认为根据这些没有使用的带宽,隐蔽通道可以作为催化剂,使用在一系列安全相关应用中。

在我们的框架中,假设有两人 Alice 和 Bob。在传送时称为 A 和 B。在 TCP/IP 协议组中使用信息隐藏来偷偷地传送信息。载体信息  $C_K$  通过普通的非理想的信道传送。这种非理想的信道通过附加的处理,将影响隐蔽信息  $C_K$  保留在这种类型的通道中。B 了解隐藏处,从而使秘密在 TCP/IP 环境中传输成为可能。

载体是网络数据包  $P_K$ 。载体是用于伪装或隐藏的载体信息。数据隐藏的目标是产生伪装网络数据包  $S_K$ ,由伪装算法产生。通过伪装的网络数据包,A 偷偷地将信息  $C_K$  发送给 B。首先产生这个包  $S_K$ (从源包  $P_K$  和  $C_K$ ),然后发送给 B。存在着一种可能性,由于安全的需要,密钥只有 A 和 B 知道。

如前面提到的,传输程序模拟,当非理想通道传送伪装的网络数据包时,将影响隐藏信息流来产生  $S_K$ ,从网络通信角度看,在数据包的顺序中,这个处理能产生位置错,因此影响了秘密信息  $C_K$ 。

另外,同时伪装网络数据包  $S_K$  可以通过中间的结点(或多个中间结点)最终到达 B。作为定义,隐蔽通道不能被这些中间结点检测到。换句话说,中间结点不能检测到伪装数据包。当处理这些包时,任何中间结点会发现  $P_K$  和  $S_K$  之间是没有区别的。

在中间结点,伪装数据包  $S_K$  由于缓冲容量的非可用性可能丢失。但是这种可能性在我们分析提出的算法中是不存在的。我们着重于网络通信量,它是不可能由于缓冲的不可用性而丢失的。在这种情况下,我们有 QoS 机制,通过它网络通信量能进行分类,并作为首

选类。另外,我们假设这有远程可能性,同时如果相关的伪装数据包被破坏,那么在传输期间会被数据链路层机制丢弃。

如果数据包  $S_K$  到达 B, 提取/检测算法应用到伪装数据包来估计隐藏的信息, 提取隐藏信息, 这种可能的影响表示为  $C_K$ 。

### 7.3.2 TCP 隐蔽通信的实现

Rowland 采用了更明确的方法, 重点在 TCP/IP 的 IP 和 TCP 头, Rowland 通过利用 IP 识别字段、TCP 初始的顺序号和确认顺序号字段设计了适当的编码和解码技术。Rowland 只提供存在的要领的证据, 同时开发 TCP/IP 协议组中隐蔽通道的存在。这项工作被认为是在这一研究领域突破性的实践。

但是, 隐蔽通信技术的不可检测性是一个问题。例如, 对 TCP/IP 头的顺序号字段进行处理, 编码方法采用的方式为: 每次使用相同字母的这种方法来偷偷通信。那么就可以将此相同的字母的编码作为同样的顺序号。

另外, 顺序号字段的使用以及确认帧字段不能被使用 ASCII 码, 因为对于具体的网络数据包, 这些字段都被认为是发送和接收数据。

网络层的包分解和重组, 也可以用于隐蔽通信, Alice 和 Bob 可以使用段偏移字段来隐藏数据位。

在传输层, 也可以使用用于路由的和错误检验的字段来隐藏信息。

TCP 协议中的信息隐藏, 它的载体是数据包。因为在网络环境中隐蔽通道指的是隐蔽信道。在 TCP 中, 我们将秘密消息根据某种嵌入算法嵌入到数据包, 生成伪数据包在信道中传输。因为伪数据包在网络中传输, 通过路由器等各种结点, 数据包可能丢失或者并没有按原顺序到达目的地, 这样我们需要重排序机制与 QoS 机制来对网络通信量进行分类。

在 TCP 中, TCP 协议的隐藏通道的基本思想是: 利用大部分防火墙和 IDS 系统的弱点, 只使用带有 ACK 标识的 TCP 包进行通信。在每个源端, 具有 ACK 标识的 TCP 报文数据域包含要执行的命令, 远程被控制端将立即发送 TCP 复位报文, 并向主控端传送命令的执行结果。从表面上整个通信过程就像是在已建立的 TCP 连接上进行的, 另外主控端和被控制端的端口分别被选为 80 端口和私有端口, 从而更加增强了通道的隐蔽性。

在 TCP 协议的隐蔽通信实现的过程中, 主要是针对 TCP 协议头。利用适当的编码技术和解码技术。TCP 协议具有信息隐藏的潜力。因为 TCP 的传输是透明的, 这就使隐蔽通信也是透明的, 并且在开放环境中的信息传输大部分都通过 Internet 数据包进行传送。

TCP 协议是在报文交换计算机通信网络中可靠的端对端的传输层协议。TCP 是面向连接的协议, 它提供了进程间可靠的通信。因为在 TCP 隐蔽通信的实现主要关注 TCP 的头。TCP 的头格式如图 7.1 所示。

根据隐蔽通道的定义可知, 隐蔽通道就是利用冗余部分。我们分析一下, 在哪些情况下 TCP 头中会出现冗余。分析 TCP 头可知, URG、ACK、PSH、RST、SYN 和 FIN 控制位是用于头控制的。对于 Urgent Pointer 字段, 只有当设置 URG 控制位时, 才用于给出紧急数

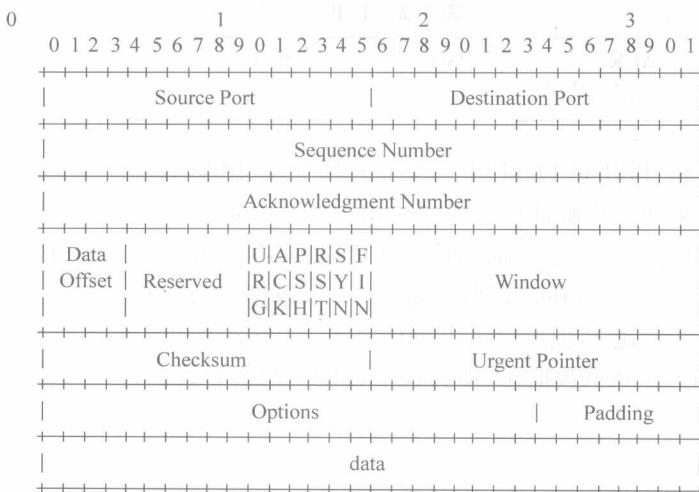


图 7.1 TCP 头格式

据的顺序号。一般而言,这个字段是不使用的,这样,我们就可以将秘密信息嵌入到此字段,Urgent Pointer 是 16 位字段,我们可以传输 8192 字节的数据。另外,因为大部分通信都需要设置 ACK 控制位,那么就可以利用 Acknowledgment Number 字段,此字段为 32 位,可以传输大约 536870912 字节的数据。在此,我们只分析了一个 TCP 数据包,而在网络中,TCP 数据数以万计,所以利用 TCP 数据包作为载体,能传输的秘密信息是前景可观的。通过以上对 TCP 头的分析,那么可以利用相应字段来实现隐蔽通信。

下面假设 Alice 和 Bob(简称为 A 和 B)之间进行隐蔽通信,Bob 的计算机可以接收到来自 Alice 发送的数据包,通过在一系列数据包中提取预先确定在特定位置的比特位或比特位块来获得秘密信息。Bob 可以很容易地重组来自于 Alice 的这些隐藏的 ASCII 码信息,A 和 B 双方使用 TCP/IP 时遵循三次握手的过程。A 想发送秘密信息“tonight”给 B。使用 SYN 包来建立和开始会话。会话的具体过程如下:

(1) A 为 SYN 包编码,A 使用 ISN(internal statement number,内码,其值为 7602176,其母对应为 t)来开始三次握手。B 的接收端口(它正在侦听并且“意识”到编码方案)使用 ISN 加参数来进行解码。得到相应的 ASCII 码,其对应的字母值为 t,也就是秘密信息的第一个字母。握手继续。在会话继续的过程中,可能出现两种情况:一种情况是,B 建立套接字来发送 SYN/ACK,并从 A 方接收到确认帧;另一种情况是,在接收到 B 的 SYN/ACK 帧之后,A 给 B 发送 RST 来放弃这次会话。

(2) 重复上述过程,A 为 tonight 的其他字母依次依据内码来分别发送相应的 SYN 包,在接收方 B,会依次分别接收到相应的字母。直到 A 给 B 发送 RST 为止。

上述是隐蔽通道使用的例子。任何网络传输所使用的传输协议都是 TCP/IP 协议。使用的信道都是合法合理的信道。第三方可能根本意识不到隐蔽通道的存在,这样,就达到了隐藏信息的目的。

那么基于这种思想,可以使用 TCP 中的序列号和确认号字段,这类的字段,对于隐蔽通信是很有用的。可以利用的冗余部分如表 7.2 所示。

表 7.2 TCP 头格式

URG	ACK	PSH	RST	SYN	FIN
0	1	1	0	0	1

另外, Alice 和 Bob 可以利用 Internet 信报控制协议 (Internet Control Message Protocol, ICMP) 来建立隐蔽通信。通过使用 source-quench 命令来完成是很容易的。当目标主机或者中间跳点不能跟上目前的传输速度时, source-quench 命令可以调整数据的传递率。这样 Alice 和 Bob 可以协商, 例如, 每第 10 个数据包可以标记为 1 或 0 比特位。为了更加精确, Bob 为了发送 1, 他可以请求重传每第 10 个数据包。在这种隐蔽通信方式中, 如果没有请求重发的第 10 个数据包, 那么这个数据包隐藏的数据为 0, 发送方 Alice 也是很清楚这一点的。

## 7.4 IGMP 中的隐蔽通信

IP 多播是允许将信息传输给多个主机, 多点传送路由器和主机为了完成多播, 必须使用 IGMP 协议来传输组成员之间的信息。这两个阶段为主机到路由器结合成一组, 另外的消息队列是路由器到主机形成一组。IGMP 封装在 IP 数据报中进行传输, IP 的目的地址是多播地址, 组群管理协议 (IGMP), IGMP 是通过充分利用 IP 堆栈来实现的。网络堆栈由不同的层构成, 每一层只和相邻的上、下层通信。IGMP 使用第 2 层和第 3 层, 使用方式与通常的单播或广播流略微不同。

在网络上传输的数据流由数据包组成, 每个数据包头均带有该数据包的起始地址和目的地址信息。单播数据流(如文件传输)的起始地址十分明显, 该地址就是 IP 地址, 位于第 3 层或 IP 数据包头中, 而起始媒体访问控制 (MAC) 地址则位于第 2 层或数据链路层上。

广播数据包的格式与单播数据包的格式一样, 但在广播数据包中, 目的地址是一个广播地址。因此, 对于网络地址为 192.168.34.0 的 IP 网络来说, 它的目的地址是 192.168.34.255。

多播数据流也必须遵循与单播和广播数据流相同的基本格式, 它们之间的差别在于其目的地址不同。IGMP 多播数据流具有一个 D 类目的地址, 范围为 224.0.0.0~239.257.257.255。该目的地址并不对应于网络中某台具体的计算机或主机, 而是与网络中距离最近的第 3 层设备相匹配, 通常为网络中的一个路由器。

当多播数据包到达路由器时, 路由器必须决定是继续传输该数据包还是停止传输。必须注意的是, 作为该数据包目的地址的 D 类 IP 地址并非某一台实际的主机, 而是一个组, 它们必须先与离它最近的路由器连接, 然后再告知数据流传输主机。如果是首次到达的数据包, 路由器便会开始“构建组”。如果其他主机没有要求路由器从该组接收数据, 那么这些数据包将被丢弃。

多播数据流请求也使用 D 类地址。如果一台主机希望寻找某个组, 它会向保留地址 224.0.0.2 发送一个“加入”信息。通过该保留地址, 此信息实际上发送给了“子网上所有的路由器”。当主机要求加入某个特殊组时, 这条路径上的路由器便会将该请求向外发送出去。最后, 当找到该组时, 数据流会顺着相同路径传回给提出请求的主机。

当主机接收完毕, 决定不再需要该数据流时, 它也向某个特殊的多播地址发送信息, 然

后该数据流便会停止发送。在实际操作中，在由各个路由器和其他第3层设备组成的不同树结构中将会“删除”这台接收完毕的主机，数据流也不再发送给它。

IGMP协议很有用，基于该协议，主机要求加入一个组的请求不必到达离数据流传输主机最近的路由器。如果一台主机申请加入数据流传输路由中的某个多播组，那么离数据流传输路由器最近的路由器便会将这些数据包进行复制，然后从这一请求多播的端口大量地向下传输给提出申请的主机。因此尽管每台提出请求的主机都可以接收到数据流，但由于这些请求并没有传输到源服务器，而数据流也只在需要多播的路由器上进行复制而不是在源服务器上复制，因此可以节省整个网络的带宽。

如果某一系统只能进行单播而不能进行多播，那么每个请求都必须返回到源服务器，然后单独从源服务器获得所需的数据流。尽管在某种意义上来说这样比较方便，例如主机可在从开始到结束的整个过程中的任一时候按自己的需要加入，但这种方法效率较低，而且并不节省网络资源。

在IP多播中，每台请求接收的PC都可以获得所需的数据流，而网络本身则管理这些PC和客户组。为了实现IGMP多播，网络必须知道数据流在何处及何时进行复制。

使用IGMP多播时，发送器（源服务器）将数据流和附加信息发送到离它最近，或在同一子网中的路由器。接收到信息后，路由器创建一个符合D类IP地址定义的组目的地址（GDA）。

路由器随后查看是否有客户机需要该多播组。如果没有，路由器便丢弃那些从发送器传来的数据包，不再继续发送。

但是，如果有客户机希望接收这些数据流（即使这一客户机位于远程网络中），路由器将执行下列步骤：

- (1) 接收器将一个专用多播IP地址发送到其子网中的所有路由器，并申明它希望加入一个多播组。
- (2) 如果子网中的路由器找到了该多播组，它开始将数据包发送给提出请求的接收器。相反，如果路由器没有找到IGMP组，它便向外发送信息并开始找寻这个组。
- (3) 通过与其他路由器通信，最初发送请求的路由器便可找寻到这个多播组。路由器之间的通信基于各种IGMP使用的“路由”协议，如多播开放最短路径优先(MOSPF)和距离向量多播路由协议(DVMRP)。
- (4) 当多播组找到后，该路径上的路由器便作为“源”路由器，发送或复制该数据流。

IGMP方案的最大好处在于节省了带宽。网络A中的远程接收器从紧接源路由器后的第一个路由器接收一个数据流。支持IGMP第2版的源路由器仅在需要复制的地方将这一数据流进行复制，而不是在源路由器进行复制，因此节省了带宽。

目前使用的IGMP版本为第2版。IGMP第1版和第2版之间的主要差别在于如何从多播组中去除客户机。第1版中规定，即使接收器不再需要某个数据流，路由器仍继续向该接收器发送数据流，并持续几分钟。在IGMP第1版中，当客户机希望停止接收数据流时，它无法告知路由器。第2版则规定，接收器可发送信息告知路由器，如果没有其他接收器出现便可停止发送数据包。因此，与第1版相比，第2版能节省更多的带宽。IGMP将存在两类消息：

- (1) 成员间的报告消息和组中主机到路由器的消息。
- (2) 成员间路由到主机的消息队列。

Handel、Sandford 和 Wolf 提出了使用协议中保留或未使用字段进行信息隐藏,这里我们提出的方法更具实践性和鲁棒性,方法是基于冗余和 IP 协议的多点传送机制。选择的头字段是灵活的,发送隐藏的数据时,并不影响标准的网络处理,如使 TCP 中的顺序号和确认号字段等,我们通过实际的仿真实验,证明是可行的。

## 7.5 IP 中的隐蔽通信

众所周知,对于分层结构的网络,IP 数据报需要封装来自于传输层的信息。例如 IP 头封装了 ICMP 消息和 IGMP 的报告和查询消息。所以 IP 头与 TCP、ICMP 和 IGMP 都有关系。在 IP 协议的分段策略中,存在着大量的冗余。IP 包头如图 7.2 和图 7.3 所示。

IPv4 包头格式				
4bit 版本号	4bit 报头长度	8bit 服务类型	16bit 数据包长度	
标识符(16bit)		标志(4bit)	分段偏移(12bit)	
生存时间(8bit)	传输协议(8bit)	报头校验和(16bit)		
源 IP 地址(32bit)			目的 IP 地址(32bit)	
选项(24bit)			填充(8bit)	

图 7.2 IPv4 包头格式

IPv6 包头格式				
4bit 版本号	4bit 优先级	24bit 流标签		
净荷长度(16bit)		下一报头(8bit)	HOP 限制(8bit)	
源 IP 地址(128bit)			目的 IP 地址(128bit)	

图 7.3 IPv6 包头格式

在 IP 中的信息隐藏也可以称为 Internet 信息隐藏,可以使用网络的任何元素和协议。下面介绍在 IP 协议中如何实现隐藏通信。

在网络传输中,假设 Alice 和 Bob 均使用 TCP/IP 协议,信息隐藏通过嵌入算法实现,然后形成伪装载体,伪装载体形成一系列的数据包序列,同时还可利用密钥对包序列加密,这个序列包通过网络传送给 Bob。为了保证安全,Alice 和 Bob 可以通过对称密钥来增加安全性,另外,对于包排序,发送方和接收方都假定执行 IPSec 协议。伪装的数据包中间要经过许多结点才能到达目的地。对于隐蔽通信,这些中间的网络结点不必检测到这些数据包,并且这里假定这些结点不能检测到并且也不会丢弃这些数据包。我们关注网络流量,可以

考虑 QoS 机制来对网络流量进行区别对待,使传送的数据包序列优先级较高而得到完整传输。这里的研究是 Handel、Sandford 和 Wolf 研究的扩展,Handel、Sandford 和 Wolf 提出使用包头中不用的字段来进行信息隐藏,在此基础上,这里加以改进,利用 Internet 协议的冗余和多点解释的处理策略。这将通过网关如防火墙或路由器等各种安全检测而不会被丢弃。用于信息隐藏头字段的选择是变化的,并且不影响数据包的正常传输和处理。在基于网络的隐蔽通信中,人们对网络数据包和协议的理解是比较含糊的,在设计的数据隐藏方案中,我们需要理解 IP 数据包及协议。在 IP 协议的分段策略中存在着冗余,在 IP 头的 Flages 字段中包含了分段信息。第一位是保留位,第二位是 DF 位,代表不分段,第三位是 MF 位,代表更多分段。未分段的数据报的分段信息均为 0,例如  $MF = 0$ ,则 13 位的段偏移=0,这就给出了冗余的条件,DF 可以携带任意的 0 或 1,数量只是受数据报最大尺寸的限制。考虑在同结构网络中的两个工作站 Alice 和 Bob,这两个用户准备通过协议组来进行隐蔽通信。假定网络的管理员已经配置好相应的安全策略。双方都知道网络的 MTU (maximum transmission unit),并且想使用分段策略,也就是任何低于或等于 MTU 的数据报都不进行分段。

在实现上述隐蔽通信中,对于所有未分段的数据报,需要 MF 字段和 fragment offset 字段值均为 0。在分段处理过程中 identification 字段不携带任何未分段数据包的任何规定范围内的值,它只需唯一的值来指明源端,目的地和协议字段即可。前者给出了冗余的条件只要知道网络所要传输的最大 MTU,则 DF 字段可以隐藏任意的 0 或者 1。

#### 数据隐藏实验 1:

假设 Alice 和 Bob 位于相同网络中的两个工作站,双方希望通过隐蔽通信来完成秘密信息的传递。并且网络管理员对安全非常关注并对网络协议如 TCP/IP 软件配置了相应安全策略。Alice 和 Bob 都知道网络中所需传输最大 MTU 的大小,并将使用分段策略来进行隐蔽通信。这里用 P1 代表网络数据包,P2 代表伪装数据包。下面给出这两个数据包。一个是正常的,一个是伪装的。伪装数据包可以会被怀疑,网络管理员可以将正常数据包与可疑数据包比较。正常数据包可以很顺利地通过网络的各种安全监控设施。从隐蔽通信的观点,正常的数据包就是恰到好处的数据包。

数据报 1 见表 7.3: 最小化的数据,最小的数据报;由于 DF 位的设置,不允许进行分段,怀疑可能由于数据报太小并且不符合结构并且不允许分段。

表 7.3 数据报 1 格式

Datagram	16 位 ID 字段	3 位 Flages 字段	13 位 Flag, offset 字段	16 位长度字段
1	XXX.. XX	010	000.....00	23

数据报 2 见表 7.4: 中等大小的数据报,由于 DF 设置,不允许分段,这种数据包比较适合信息隐藏。

表 7.4 数据报 2 格式

Datagram	16 位 ID 字段	3 位 Flages 字段	13 位 Flag, offset 字段	16 位长度字段
2	XXX.. XX	010	000.....00	474

数据报 3 见表 7.5：中等大小的数据报，由于 DF 位没有设定，所以允许分段，但是由于 Alice 和 Bob 所要进行的隐蔽通信的数据包都不会超过最大的 MTU，所以发送方和接收方的数据包不需要进行分段。

表 7.5 数据报 3 格式

Datagram	16 位 ID 字段	3 位 Flages 字段	13 位 Flag. offset 字段	16 位长度字段
3	XXX..XX	000	000.....00	21

所以当 Alice 需要单独传输 1 和 0 给 Bob 时，DF 位可以是 010 和 000，但这种通信中的限制是必须知道最大 MTU。并且 Alice 和 Bob 的通信不能太过频繁，如果过于频繁，将会引起网络管理员的怀疑。

#### 隐蔽通信实验 2：

在第一组实验中，数据报 2 和数据报 3 可以实现隐藏通信，那么考虑其头字段，如 Id 字段。如果在实验 1 的数据报 2 或 3 中综合使用 ID 字段，在这种综合应用中，规则是对伪装的数据包具有唯一的独特的 ID 字段。这样，Alice 和 Bob 就可以传送大量的隐藏信息了。也就是说，这个实验中，是将 DF 字段和 ID 字段结合在一起使用。通过恰当的嵌入算法，很容易地实现了隐藏通信。

另外，16 位的 ID 字段有 65536 个特定的值，所以 Alice 和 Bob 可以多次进行隐藏通信而不引起怀疑。所以这种隐藏通信中可以从一对一扩展到一对多，但局限性仍在于必须知道网络的最大 MTU 和必须位于相同的网络之中。

#### 信息隐藏实验 3：

在这个实验中，不再需要知道网络中的最大 MTU，但在 IP 头中要求没有 options 选项，在 Internet 通信中，经常不使用这个选项，并且许多安全分析并不考虑这个选项。这就提供了很好的机会。如果没有这个选项，那么头字段的版本号的值为 4(0100)，Internet 头字段长度字段为 5。在这次实验中，是标志字段和版本字段与 Internet 字段的结合。发送方 Alice 需要用 XOR 操作进行编码，具体的过程为：首先，因为版本号为 0100，Internet 字段为 0101，这里共有 8 位，让我们设序列为  $[h_1, h_2, \dots, h_8]$ 。标志字段记为  $[i_1, i_2, \dots, i_{16}]$ ，前 8 位的字段为  $[h_1, h_2, \dots, h_8] = 01000101$ ，接下来的标志字段的后 8 位记作  $[c_1, c_2, \dots, c_8]$ ，可以用来隐藏信息，三者之间的关系为： $i_1 = h_1 \text{xor } c_1$ 。在此实验中，选择隐藏信息的比特位，然后通过与版本前八位的异或操作得到标志位来形成新的标志字段用于隐藏通信。在这里，假设传字母 A，它的 ASCII 码值为 65，二进制值为 01000001，所以  $[c_1, c_2, \dots, c_8] = [01000001]$ 。与版本号异或操作为： $[01000101] \text{xor } [01000001] = [00000100]$  所得结果作为标志位。数据报 4 格式如表 7.6 所示。

表 7.6 数据报 4 格式

4 位版本号 0100	4 位 IHL 0101	8 位 TOS XXXXXXUU	16 位 Tot. Len XXXXXXXXXXXXXXXX
16 位 Ident 0000 0100 RRRRRRRR		.....	.....

这样通过标志位与版本位的结合使用,通过异或操作,我们就可以进行隐蔽通信。在接收方 Bob 处,他能得到版本位和标志位字段,为了得到隐蔽的信息,他也只需要将版本位与标志位进行异或操作: $[01000101] \text{ xor } [00000100] = [01000001]$ ,即得到 A 字母的二进制值。虽然对 IP 头中标志字段进行处理,但这些经过处理的数据包都能顺利地通过防火墙等安全隔离设施。因为标志位的后八位使用的是随机序列,所以这种算法能很好地实现隐蔽通信。

上面分析了Ipv4 中的隐蔽通信,在Ipv6 中同样适用。

## 7.6 本 章 小 结

在本章中,我们分析了在网络环境中隐蔽通信的存在,并用实验证明了可以通过传输层和网络层协议的冗余机制来实现。主要是对包头进行处理,这里分析的协议有TCP/IP,IGMP 和 ICMP。主要是前者,提出了三种方案来实现隐蔽通信。

这项工作基于网络环境进行,而目前大部分的通信都必须经过网络,所以这是一个网络安全与信息隐藏相结合的一个方向,具有很广阔前景,在这里,值得进一步研究的是在Ipv6 环境下的信息隐藏的实现,如包排序和包重组策略中来实现信息隐藏等。

## 7.7 复 习 题

- 7.1 隐蔽通信是什么?
- 7.2 隐蔽通道是什么?是如何分类的?
- 7.3 谈谈自己理解的 TCP/IP。
- 7.4 在 TCP 中如何实现隐蔽通信?
- 7.5 在 IP 中如何实现隐蔽通信?
- 7.6 思考在目前的 IPv6 环境中如何实现隐蔽通信。

# 第8章

## 隐写分析技术

### 本章目标

- 理解隐写分析技术。
- 了解隐写技术的分类。
- 理解隐写分析的评价指标。
- 理解隐写分析通用原型系统及相应的算法。

随着隐写技术的迅速发展,针对隐写这一隐蔽的通信手段,反隐写技术也相应地发展起来,其中隐写分析就是利用各种统计分析手段对隐写技术进行攻击,目的是检测载体中秘密信息的存在从而阻断隐蔽通信的进行。本章主要阐述隐写分析技术的定义、隐写分析分类、隐写分析算法的有效性评价和隐写分析的原型系统,并分析典型的隐写分析算法。

### 8.1 隐写分析概述

在本节中介绍隐写分析的定义与分类。

#### 8.1.1 隐写分析定义

隐写分析(steganalysis)技术是对表面正常的图像、音频、视频等媒体信号(尤其是通过互联网进行传输的信号)进行检测,判断其中是否嵌有秘密信息(这些秘密信息是通过一定的隐写算法嵌入的),甚至只是指出媒体中存在秘密信息的可能性,这样就可以找到敌对隐蔽通信的信源,从而阻断隐蔽通信的信道。

由于隐写者必须通过修改原始数据才能实现秘密信息的嵌入,因此载体数据的统计特性不可避免地会发生一些变化。虽然分析者并不知道原始载体,但可以利用载体数据的统计特性的异常来觉察到秘密信息的存在。即使不能破解秘密信息的具体内容,分析者仍可以阻断隐蔽通信并追查秘密信息的收发双方。

#### 8.1.2 隐写分析分类

从攻击的角度,我们用阐述隐写术的“囚犯”问题来对隐写分析进行分类。

### 1. 被动攻击

进行秘密通信的囚犯的来往信件都要经过看守的检查,看守检查信件后判断是否存在秘密消息并作不同的处理,称为被动攻击。

被动攻击根据目标和条件的不同,可分为以下几种情况:

- (1) 分析者已知隐写算法并同时持有原始载体和含密载体对象(known stego attack)。
- (2) 分析者知道含有隐秘的信息或它的某种派生形式(known message attack)。
- (3) 将已知原始媒体与待分析含密对象比较,检测其中是否存在差异(known cover attack)。
- (4) 在已知敌方所用隐写工具和隐写内容的情况下对待检测载体进行检测(chosen stego attack)。
- (5) 分析者可以使用某种隐写工具嵌入选择的消息产生含密对象,以确定其中是否涉及某一隐写工具或隐写算法的相应模式(chosen message attack)。
- (6) 分析者仅仅持有可能含密的载体对象,对隐写内容和有可能使用的隐写算法完全不知,是完全的盲分析。

上述最后一种情况在技术上最具有挑战性,是隐写分析的重要研究内容,毫无疑问,它是隐写分析的终极目标,即成功地实现针对任何对象、任何隐写方法的完全的盲分析。然而对隐写算法和隐写内容完全不知的隐写分析往往非常困难,因此,针对一些有效的隐写方法和特定的载体对象研究有针对性的分析技术具有重要意义。例如,LSB 隐写算法由于简单、性能好而被广泛使用;JPEG 和 GIF 格式的图像因为易于传输经常被作为隐写的载体。因此,针对这些隐写算法和载体的隐写分析技术具有重要的实际意义和应用价值。

### 2. 主动攻击

主动攻击又叫做积极攻击,如果看守不经过判断就对消息进行修改的攻击称为主动攻击。例如在传输链路的某些环节设置所谓的主动卫士(active warden),其原理就是对通过的所有媒体信息进行某种处理,既不对信号产生任何可察觉的损伤,又使其中可能存在的隐蔽信息遭到破坏而无法提取,达到阻断隐蔽通信的目的。因此,所谓主动攻击,就是分析者直接在数字媒体中广泛引入干扰使得载体数据中可能存在的秘密信息无法提取而并不分析某一数字媒体中是否含有秘密信息。这类攻击要满足两个条件,即引入的干扰不能影响媒体的正常使用,同时不应该暴露积极攻击行为本身,亦即要用尽可能弱的干扰对秘密信息造成尽可能强的损伤。

### 3. 隐写分析其他内容

除了上述的被动攻击和主动攻击外,隐写分析还包括估计嵌入信息量的多少,即对待检测载体,不仅要检测秘密信息的存在性,如果存在秘密信息,还要估计嵌入秘密信息的数量。更进一步,在隐写分析的基础上提取秘密信息,但在未知隐写算法和密钥的情况下还难以解决。

## 8.2 隐写分析评价指标

对于隐写分析技术的评价,这里仅讨论被动隐写分析方法的评价,可以采用4个评价指标:准确性、适用性、实用性和复杂度。

### 1. 准确性

准确性指检测的准确程度,是隐写分析最重要的一个评价指标,一般采用虚警率和检测率表示,两个指标的关系可以描绘成如图8.1所示的检测器接收操作特性(detector's receiver operating characteristic, ROC)二维平面。虚警率是把非隐藏信息误判为隐藏信息的概率,表示为: $\alpha=P(\text{隐藏信息}|\text{非隐藏信息})$ ;检测率是把隐藏信息正确判为隐藏信息的概率,表示为: $\beta=P(\text{隐藏信息}|\text{隐藏信息})$ 。此外,还需要考虑漏报率,即把隐藏信息错误判为非隐藏信息的概率,表示为: $\eta=1-\beta=P(\text{非隐藏信息}|\text{隐藏信息})$ 。

隐写分析要求在尽量减少虚警率和漏报率的前提下取得最佳检测率。在虚警率和漏报率的减少无法兼顾的情况下,首先减少漏报率。

利用以上指标,可以得出全面衡量隐写分析准确性的指标全局检测率 $P_r=1-P_e$ ,其中 $P_e$ 为平均错误概率:

$$P_e = (1-\beta)\rho + \alpha(1-\rho) = \eta\rho + \alpha(1-\rho)$$

当 $\alpha=\beta$ 即点 $(\alpha, \beta)$ 落在图的45°对角线上时,全局检测率为50%,属于随机猜测,也即瞎猜,此时隐写分析检测器无效。当全局检测率达到85%以上,可以认为检测器性能良好。

### 2. 适用性

适用性指检测算法对不同嵌入算法的有效性,可由检测算法能够有效检测多少种、多少类隐写算法或嵌入算法来衡量。

### 3. 实用性

实用性指检测算法可实际推广应用的程度,可由现实条件是否允许、检测结果是否稳定、自动化程度的高低和实时性等来衡量。其中实时性可以用隐写分析算法进行一次隐写分析所用时间来衡量,用时越短则实时性越好。

### 4. 复杂度

复杂度是针对隐写分析算法本身而言的,可由隐写分析算法实现所需要的资源开销、软硬件条件等来衡量。

到目前为止,还没有人给出准确性、适用性、实用性和复杂度的定量度量,只能通过比较不同检测算法之间的实现情况和检测效果得出一个相对的结论。

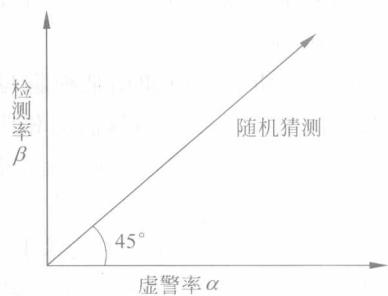


图8.1 检测器ROC平面

### 8.3 隐写分析通用原型系统

图 8.2 给出了图像隐写分析通用原型系统。隐藏秘密消息后的载体图像称为隐藏图像。将待检测图像输入后进行特征提取,根据图像的特征是否被改变以及改变的程度来判别图像是否隐藏了秘密消息。

特征提取包括特征寻找与特征选择。根据特征提取与嵌入算法的关系,图像隐写分析有两条途径。一是针对某种具体的嵌入方法提取其专有特征,根据这些专有特征进行判别,叫做专用隐写分析技术;二是寻找独立于具体的嵌入算法之外的通用特征,根据这些特征进行判别,叫做通用隐写分析技术。专用隐写分析技术可以准确检测采用特定嵌入方法的隐藏图像,准确性高但适用性低。通用隐写分析技术的准确性不如专用隐写分析技术高,但适用性高。因此,无论专用隐写分析还是通用隐写分析,寻找对信息隐藏敏感的特征是隐写分析实现的关键。判别是根据提取的特征对图像归类。修正是根据判别结果的好坏对提取的特征以及判别系数或阈值作调整,以提高判别的准确性。这里秘密消息的提取包括估计秘密消息的嵌入量和秘密消息的嵌入位置而不是提取秘密信息本身。虽然该过程的实现难度更大,但是已经有若干检测算法在检测是否存在秘密消息的同时,可以比较准确地估计出秘密消息的嵌入量。目前,仍然没有能够准确确定秘密消息的嵌入位置的研究报道。成功破译是在秘密消息提取后的解密工作,到目前为止还没有隐写分析成功破译的报道。

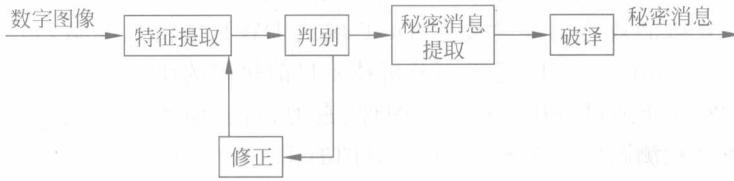


图 8.2 图像隐写分析通用原型

### 8.4 隐写分析算法

隐写分析算法根据提取的特征的不同,可以分为专用隐写分析和通用隐写分析;根据特征所在域的不同,又可以分为空域隐写分析和频率域隐写分析。下面首先简要介绍有关隐写分析算法,然后具体分析一种隐写分析算法,以抛砖引玉。

#### 8.4.1 专用隐写分析算法介绍

空域隐写分析算法的攻击对象主要是空域最低有效位信息隐藏,包括 EzStego、S-Tools、Stash、Steghide、Gifshuffle、Stegno、BPCS 等隐写算法,是隐写分析技术早期热度比较高的研究内容。

Westfeld 于 1999 年采用了 Chi-square 统计量统计调色板图像嵌入秘密消息前后出现相似颜色对的概率比,这种方法能够准确检测连续嵌入秘密消息的调色板图像,但对于随机

嵌入秘密消息的图像的检测无效。

Westfeld 于 1999 年使用 Chi-square 统计量统计颜色频度,能够检测 Jsteg 连续嵌入秘密消息的 JPEG 格式图像,对于消息嵌入量较大的情况检测准确率较高。但是该方法对离散嵌入情况的检测无效。

Fridrich 于 2001 年提出的 RS(regular groups and singular groups) 隐写分析方法把图像像素分成规则类、异常类和不可使用类,根据待测图像 LSB 置换操作前后各类像素组的变化曲线能够可靠地检测灰度和真彩色图像并估计秘密消息的嵌入量,但 RS 的检测结果直接受载体图像噪声、随机性和秘密信息嵌入位置的影响。

Dumitrescu 于 2003 年提出的样本对分析法达到了与 RS 最优检测等效的结果。该分析法根据相邻像素值的奇偶性质将像素对分为 4 种基本集合,秘密消息的嵌入导致像素对从一个集合转换到另一个集合,根据集合更改的比例采用二次方程建模来估计嵌入量。该方法适用于对连续信号采样的检测,但检测结果直接受秘密信息嵌入位置影响,对非随机嵌入无效。

张涛于 2004 年提出图像差分直方图的转移系数作为 LSB 平面与其余位平面之间的弱相关性度量,并以此为基础构造原始图像与隐藏图像的分类器。在嵌入量较大的情况下该算法检测效果优于 RS 隐写分析,但检测效果受秘密消息嵌入位置和随机性的影响。

可见,空域隐写分析算法较多地围绕颜色对现象展开研究,方法经历了从简单分析隐藏图像颜色对到采用比较复杂的实验手段(如再次嵌入秘密消息、归类、划分集合等)来获得颜色对变化量的过程,这些方法的原理也可以用到变换域的隐写分析中。

变换域隐写分析的攻击对象主要是 DCT 域或 DWT 域信息隐藏,包括 JSteg、Jsteg-Shell、JPHide、F5、Outguess、MB,是隐写分析技术目前热门的研究内容。

Fridrich 于 2002 年通过解压缩待检测图像、裁剪、再压缩等步骤估计载体图像的 DCT 系数直方图,根据待检测图像直方图和估计直方图的相关改变量估计 F5 隐写算法的秘密消息嵌入量。该方法能准确检测最低 10% 的嵌入量,但对于具有特殊网格结构的图像检测无效。

Fridrich 于 2002 年对待测图像进行 Outguess 嵌入操作,根据载体图像与隐藏图像像素块边界的增量差来估计隐写算法 Outguess 的嵌入量。该方法不需要阈值,对不能由嵌入秘密消息的长度估计图像的宏观改变量的情况,以及对以 DCT 系数的增减量做嵌入算法的情况无效。

可见,DCT 域隐写分析主要围绕 DCT 系数的统计特性及其对空域像素的影响进行研究,包括了对载体图像 DCT 系数的估计及空域像素块不连续性的计算。研究的方法经历了从简单的一阶统计分析到采用比较复杂的实验手段来获得相关变化量的过程,总体来说适用面较窄,实用性不高,有待于进一步研究。

DWT 域隐写分析的研究报道较少,Shaohui Liu 于 2004 年针对 DWT 域 QIM 嵌入算法,提出了基于 DFT 域能量差分的检测算法,检测率达到 90%。该文是检测 DWT 域隐写术的有益尝试。

#### 8.4.2 通用隐写分析算法介绍

Avcibas 于 2003 年提出的 IQMs(image quality metrics) 方法,采用变量分析技术来分

析和选取可用于区分载体图像和隐藏图像的质量度量,根据所选取的图像质量特征采用多元回归方法对图像进行分类。该方法对多种隐写术的检测有效,但是需要对分类器进行训练,性能一般。

Farid 于 2004 年采用 QFM 分析图像小波域系数及其预测误差的高阶统计量,再分别采用 Fisher 线性判别式、线性与非线性支持矢量机来判别和归类的方法,对 DCT 域隐写算法和以自然图像为载体的隐写算法效果较好。该方法需要对分类器进行训练,对嵌入量低的空域隐写方法和 OutGuess 隐写算法的检测无效。

国内学者对通用隐写分析方法也进行了探索。提出了一种基于直方图频域统计矩的图像通用隐写分析技术,该方法以图像小波子带系数直方图频域统计绝对矩作为特征,通过分类器进行分类,以区分原始图像和载密图像,取得了较好的效果。但该方法也需要对分类器进行训练。

可见,通用隐写分析主要围绕嵌入秘密消息前后待检测图像的总体、局部、相关等特征值及具有训练模式的判别方法进行研究,但是通用特征的选取和阈值的确定非常困难,而且复杂度偏高,实用性不强,准确性较低,无法控制虚警率和漏报率,无法估计信息隐藏量。

### 8.4.3 GPC 隐写分析法

对于绝大多数图像而言,采样点之间是具有较强的相关性的,而秘密信息通常经过压缩或加密,不具有相关性,GPC 分析法就利用相邻像素的相关性对隐写行为进行检测,我们对该算法进行了一定的改进和完善,提高了检测率,并通过二次嵌入建模来估计秘密信息的嵌入量。具体方法如下所述。

#### (1) 隐写分析算法

首先将载体图像数据的像素集对应于三维空间的一个离散点集,即将像素的位置对应于 XY 平面上的一点,而将其灰度值对应于 Z 轴上的一个值,再将相邻像素用直线连接起来,就可以得到三维空间中的起伏网格,类似渔网。图 8.3 是图像 Lena(如图 8.4 所示)中的  $30 \times 30$  个像素的局部网格图,X,Y 代表像素的位置,Z 代表灰度值。

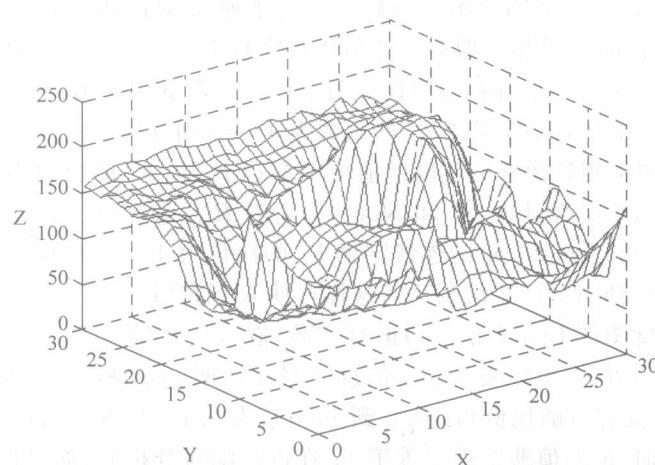


图 8.3 图像 Lena 的局部网格图



图 8.4 图像 Lena

然后建立两个平行于 XY 平面的平面簇,平面簇  $P_0$  由  $z=1.5, z=3.5, z=5.5, \dots, z=255.5$  组成,平面簇  $P_1$  由  $z=0.5, z=2.5, z=4.5, \dots, z=254.5$  组成,这样相邻像素之间的连线会穿过平面簇  $P_0$  和  $P_1$ ;令网格图中的网格线穿过平面簇  $P_0$  和  $P_1$  的次数分别为  $N_0$  和  $N_1$ 。如果载体图像是自然图像,根据统计特性则认为  $N_0 \approx N_1$ ;反之,若载体图像经过 LSB 隐写,由于隐写仅替换最低比特位,相邻像素的网格线不会跨越  $P_0$  中的平面,只会跨越  $P_1$  中的平面,所以  $N_0$  保持不变,  $N_1$  会增大。

GPC 分析的基本原理可以总结为:若载体图像是自然图像,由于相邻像素之间的相关性,  $N_0 \approx N_1$ ;若载体图像经过了 LSB 隐写,则会减弱相邻像素之间的相关性,从而使  $N_1$  增大。最终通过比较  $N_0$  和  $N_1$  来判断秘密信息的有无。

## (2) 隐写分析算法改进

根据以上基本原理,可通过比较  $N_0$  和  $N_1$  的大小判断秘密信息的有无。令  $R=N_1/N_0$ , 定义阈值  $T$ ,当  $R>T$  时,认为该载体数据含有秘密信息。

在现实应用中,因为  $N_0$  和  $N_1$  的值较大,  $R$  的值不会对  $N_0$  和  $N_1$  的变化太敏感,即  $R$  的值对隐写过程不敏感,从而阈值不容易确定。为了使  $R$  对嵌入信息比较敏感,可以采用如下改进策略:一方面,如果相邻像素的差值使得穿越  $P_0$  和  $P_1$  的次数相等,则穿越的次数不计入  $N_0$  和  $N_1$ ,或各记 1 次;例如,假设有两个相邻像素  $p_1(22)$  和  $p_2(112)$ ,那么  $p_1$  和  $p_2$  的连线穿越  $P_0$  的次数为 45 次,穿越  $P_1$  的次数为 45 次,则这两个次数由于相等不计入  $N_0$  和  $N_1$ ;另一方面,如果相邻像素的灰度差值使得穿越  $P_0$  和  $P_1$  的次数不相等,那么这一对像素对应的穿越次数不全部计入  $N_0$  和  $N_1$ ,这样可使  $R$  比较敏感。例如,假设两个相邻像素  $p_1(22)$  和  $p_2(111)$ ,则  $p_1$  和  $p_2$  的连线穿越  $P_0$  的次数为 44 次,穿越  $P_1$  的次数为 45 次,我们可以分别计 1 次到  $N_0$ ,计 2 次到  $N_1$ ,这样可使  $R$  保持敏感。

对标准测试灰度图像 Lena(图 8.4)进行实验,嵌入信息量与  $R$  的关系如图 8.5 所示。图 8.5 中纵坐标表示  $R$ ,横坐标表示秘密信息的嵌入比例  $\alpha$ ( $\alpha$  是嵌入的秘密信息的长度和载体图像的 LSB 最大容量的比例), $\alpha$  从 0 到 1,步长为 0.1。从图中可以看出, $\alpha$  等于 0 时,即无秘密信息嵌入时, $R$  的值非常近似等于 1;在进行隐写分析时,如果待检测图像的  $R$  值明显大于 1,则认为图像中含有秘密信息。可以通过实验来确定阈值。

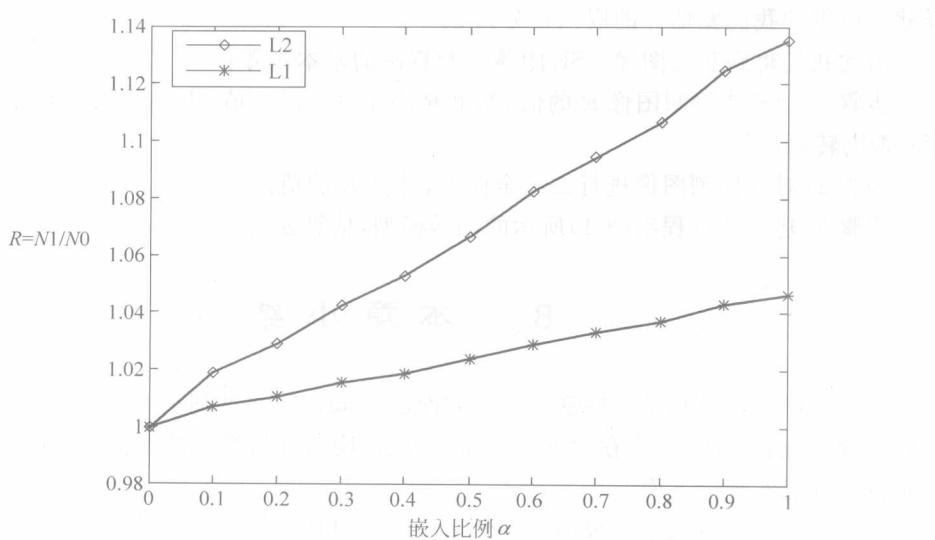
图 8.5 嵌入信息量与参数  $R$  的关系

图 8.5 中  $L_1$  表示没有采取策略时的  $\alpha$  和  $R$  的关系曲线,  $R$  对  $\alpha$  的变化不敏感, 表现为曲线比较平缓;  $L_2$  表示采取上述策略后  $\alpha$  和  $R$  的关系曲线,  $R$  对  $\alpha$  的变化比较敏感, 表现为曲线比较陡峭。在相同的阈值情况下, 采取策略后, 可以降低漏检概率和虚警概率, 随后通过实验可以加以说明。

### (3) 信息隐藏量估计

观察图 8.5 中的嵌入量  $\alpha$  和  $R$  的关系曲线, 当  $\alpha$  以步长 0.1 增加时,  $R$  的值近似呈线性增长, 在图中表现为一条直线。

然后用下面的方法估计隐写信息量  $\alpha$ 。图 8.5 中  $\alpha$  和  $R$  的函数关系近似一次函数, 我们选择用一次函数  $y=a+b\alpha$  来拟合。实验证明, 一次线性函数可以很好地对这一函数关系建模。下面通过两个关键点(用  $(\alpha, y(\alpha))$  来表示)来估计嵌入比例  $\alpha$ 。

两个关键点是零嵌入  $(0, y(0))$  情况和全嵌入  $(1, y(1))$  情况。对于零嵌入情况, 通过大量实验, 结果表明当  $\alpha$  等于 0 时,  $R$  分布在 1 附近, 这里我们经过实验表明, 当  $\alpha=0$  时, 选  $R=1$  建模的效果较好。

对于全嵌入情况, 对待检测图像进行二次隐写, 嵌入率  $\beta$  为 100%。经二次隐写后, 有部分像素经历了两次改变又回到原始的灰度值, 而在两次隐写中仅改变过一次灰度值的像素占全部像素的  $(\alpha+\beta-\alpha\beta)/2$ , 即相当于经历了一次嵌入率为  $(\alpha+\beta-\alpha\beta)$  的隐写。也就是说, 当二次嵌入率为 100% 时, 仅经历过一次灰度值变化的像素正好占全部像素的 50%, 也就相当于经历了一次嵌入率为 100% 的隐写。因此, 可以用二次全嵌入近似一次全嵌入时的情况。

由此我们可以得到方程组:

$$\begin{cases} y(0) = a + b \times 0 \\ y(\alpha) = a + b \times \alpha \\ y(1) = a + b \times 1 \end{cases} \quad (8-1)$$

在方程组(8-2)中,  $y(0) \approx 1$  并且  $y(1)$  的值可以用二次全嵌入时所得到  $R$  的值近似, 这

样就可以得到我们要估计的嵌入百分比  $\alpha$ 。

由此我们得到灰度图像 LSB 隐藏检测算法的基本步骤：

步骤 1, 计算待检测图像  $R$  的值, 如果  $R$  小于规定的阈值, 则认为图像不含秘密信息, 结束; 否则转入步骤 2。

步骤 2, 对待检测图像进行二次全嵌入, 计算  $R$  的值。

步骤 3, 建立如方程组(8-1)所示的函数模型, 估算  $\alpha$  的值。

## 8.5 本章小结

国内外研究人员和学者在隐写分析方面已经取得了一些研究成果, 但是仍有不少问题需要继续研究和解决, 集中在隐写分析研究方法、隐写分析算法的评价和隐写分析理论构建与实用系统的实现三方面。

目前隐写分析研究主要采用统计分析方法。但是近年来出现了抗统计分析的信息隐藏方法, 可以做到在嵌入秘密消息的同时保持载体的统计特征不变。这给采用统计分析方法的隐写分析带来了新的挑战。

隐写分析方法的评价方面还没有形成十分有效的隐写分析评价标准, 有必要建立用于检测的测试图像库和相关的一系列评价量与评价手段, 文献[5]在隐写分析的评价方面进行了有益的探索。

隐写分析理论构建方面把隐写分析简化为检测载体的噪声。那么如何区分随机噪声和秘密消息是一个亟待解决的问题。鉴于检测准确性、实用性和适用性等各方面的综合要求, 将统计分析和归类判断的方法相结合, 实现全自动检测, 是构建实用隐写检测系统的研究方向。

隐写分析技术和隐写技术是对立统一的关系, 新的隐写方法不断被提出, 有些方法不久就被研究人员找到检测的办法, 这个结果又推动新的隐写方法的提出, 这种交替更新推动了信息隐藏技术不断向前发展。

## 8.6 复习题

- 8.1 隐写分析技术是什么?
- 8.2 隐写分析是如何分类的?
- 8.3 谈谈自己对隐写分析评价指标的理解。
- 8.4 隐写分析的通用原型都有哪些? 你是如何理解的?
- 8.5 隐写分析算法都有哪些类型? 各自的优缺点是什么?
- 8.6 什么是 GPC 隐写分析法?

## 参 考 文 献

1. The Oxford English Dictionary (corrected reissue). Oxford, U. K. : Clarendon, 1933.
2. A Tacticus. How to Survive Under Siege/Aineias the Tactician (Clarendon Ancient History Series). Oxford, U. K. : Clarendon, 1990, 84~90.
3. Herodotus. The histories. London. English: J. M. Dentsons, Ltd, 1992.
4. M K Mih, Cak, R Venkatesan. Blind Image Watermarking Via Derivation and Quantization of Robust Semi-Global Statistics. Proc. IEEE ICASSP 2002, Orlando, FL, May 2002.
5. B Chen, G W Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Information Theory, 2001, vol. 47. 1423~1443.
6. M K Mih, Cak, P Moulin. Information Embedding Codes Matched to Locally Stationary Gaussian Image Models. Proc. IEEE ICIP 2002, Rochester, NY, Sep. 2002.
7. M K Mih, Cak, R Venkatesan, M Kesal. Cryptanalysis of Discrete-Sequence Spread Spectrum Watermarks. Proceedings of 5th Information Hiding Workshop, Holland, Oct, 2002.
8. J R Hernandez, J M Rodriguez, F. Perez-Gonzalez. Improving the performance of spatial watermarking of images using channel coding. Signal Processing 2000, 1261~1279.
9. S Pereira, S Voloshynovskiy, T Pun. Optimal transform domain watermark embedding via linear programming. Signal Processing, 2001, 1251~1260.
10. F Perez-Gonzalez, J R Hernandez, F Balado. Approaching the capacity limit in image watermarking: a perspective on coding techniques for data hiding applications. Signal Processing, 2001, 1215~1238.
11. R Baitello, et al. From watermark detection to watermark decoding: a PPM approach. Signal Processing, 2001, 1261~1271.
12. S Voloshynovskiy, S Pereira, T Pun. Attacks on digital watermarks: classification, estimation-based attacks, and benchmarks, IEEE Communications Magazine, 2001, 118~126.
13. M Barni, et al. Watermark embedding: hiding a signal within a cover image. IEEE Communications Magazine, 2001, 102~108.
14. R C Gonzalez, R E Woods. Digital Image Processing, Upper Saddle River, New Jersey, Prentice Hall, Inc. ,2002.
15. J R Hernandez, M Amado, F Perez-Gonzalez. DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure. in IEEE Trans. Image Processing, 2000, vol. 9. 55~68.
16. P Meerwald, A Uhl. Watermark Security via Wavelet Filter Parameterization. Internation Conference on Image Processing, Thessaloniki, Greece, 2001.
17. P Meerwald, A Uhl. A Survey of Wavelet-Domain Watermarking Algorithms. EI San Jose, CA, USA, 2001.
18. Fridrich J, Goljan M, Du R. Detecting LSB Steganography in color and Gray-Scale Images. Magazine of IEEE Multimedia, Special Issue on Security, 2001. Issue: 22~28.
19. Fridrich J, Goljan M. Practical Steganalysis of Digital Images-State of the Art. In: Security and Watermarking of Multimedia Content IV, Proceedings of SPIE, 4675. USA: San Jose. Jan. 2002, 1~13.
20. Avcibas I, Memon N, Sankur B. Steganalysis using image quality metrics. IEEE Transactions on Image Processing, 2003, 221~229.

21. Lyu S, Farid H. Steganalysis using colour wavelet statistics and one-class support vector machines. SPIE Symposium on Electronic Imaging, San Jose CA, 2004.
22. Wang H, Wang S. Cyber Warfare-Steganography vs. Steganalysis. Communication of the ACM, 2004, 76~82.
23. Moulin P. The Role of Information Theory in Watermarking and its Application to Image Watermarking. Sigal Processing, 2001, 1121~1139.
24. Dumitrescu S, Wu X L, Wang Z. Detection of LSB steganography via sample pair analysis. IEEE Transactions on Signal Processing, 2003, 51(7): 1995~2007.
25. Fridrich J, Goljan M, Hogea D. Steganalysis of JPEG Image: Breaking the F5 Algorithm. In 5th International Workshop on Information Hiding, Lecture Notes in Computer Science, 2578. Springer-Verlag, 2002, 310~323.
26. Liu S H, Yao H X, Gao W. Steganalysis of data hiding techniques in wavelet domain. International Conference on Information Technology: Coding Computing, ITCC. Las Vegas USA, 2004, (1): 751~754.
27. Avcibas I, Memon N, Sankur B. Steganalysis using image quality metrics. IEEE Transactions on Image Processing, 2003, 12(2): 221~229.
28. Lyu S, Farid H. Steganalysis using colour wavelet statistics and one-class support vector machines. SPIE Symposium on Electronic Imaging, San Jose CA, 2004.

- \* 详细阐述了信息隐藏的基本原理
- \* 分析了隐写术与数字水印的典型算法并给出了实例及代码
- \* 探讨了隐蔽通信，并进行了相应的实验

ISBN 978-7-302-18324-2



9 787302 183242 >

定价：18.00元