中文摘要

视频传输以及视频监控作为信息化建设中一个主要的领域，得到了广泛的研究。由于网络的公开性以及其他不安全因素，我们往往希望能够将我们要传送的数据，图像以及视频等信息进行有效的加密传输。现有的视频加密传输系统大都采用传统密码学中的加密方案进行软件加密并网络传输，而视频实时监控也已经从传统模拟集中监控方式转变为数字化视频监控。混沌以其对初始条件和系统参数的敏感性，生成的密钥流满足传统密码系统的扩散与混淆原则的特性，越来越广泛的应用到密码学领域。本论文将主要研究混沌流密码以及自同步流密码设计方案，旨在将其应用到视频传输以及视频监控中来，构建一个实时视频加密传输系统，对视频加密领域有较高应用价值。

本文首先研究了序列密码以及混沌理论，对混沌方程的特性（倍周期分叉、混沌吸引子、Lyapunov 指数、Kolmogorov 熵）进行了仿真。然后研究了混沌同步保密通信理论，对混沌遮掩以及混沌键控方法进行了研究与仿真分析。接下来就数字化混沌密码理论进行了研究与分析，完成了混沌序列密码算法以及自同步流密码算法的硬件实现。最后通过将加密算法下载到网络加密卡中，利用视频采集卡 SDK 二次开发完成了服务器端与客户端应用程序软件开发，构建了视频加密传输系统。

本文通过对混沌理论的研究，对混沌应用于密码学中的两个方向分别进行了仿真和实现，设计了基于混沌算法的加密方法以及自同步流密码方法。论文的研究工作将对数字化混沌密码的应用提供参考，并将为数字化视频加密传输与监控系统提供一种可行方案。

关键词：混沌；视频加密传输；混沌同步；自同步流密码

Abstract

As one of the main field in informatization construction, video transmission and video monitoring have been widely studied. Because of the network open and other insecurity factors, we often want to transmit data, image and video information by effective encryption. The existing video encryption transmission systems mostly use traditional cryptography on network transmission by software encryption scheme, and real-time video monitoring has also changed from traditional analog centralized monitoring mode to digital video monitoring. Since chaos has the following characteristics: first, sensitivity to the initial conditions and system parameters; second, the key stream satisfies diffusion and confusing principles of traditional cryptography.It has been applied in cryptography more widely. This thesis mainly studies chaotic stream cipher and self-synchronizing scheme, aims to apply in the video transmission and video monitoring to construct a real-time video encryption transmission system, which has a higher application value in video encryption area.

This dissertation first studies sequence cipher and chaos theory, simulates the characteristics of chaotic equation (the period-doubling bifurcation, chaotic attractor, Lyapunov exponent, Kolmogorov entropy) and chaotic synchronization secure communication theory, in which Chaos Masking and Chaos Shift Keying are studied and simulated.Then we study and analyze the digital chaotic cryptography, complete hardware realization of the chaotic sequence cryptographic algorithm and self-synchronization stream sequence cryptographic algorithm. Finally, we download encryption algorithm to the network encryption card, use secondary development of video capture card's SDK (software

development kit) to complete the server and the client application software development, and construct the video encryption transmission system.

Based on the study of chaos theory, in this thesis, we simulate the two directions of chaos applied in cryptography respectively, and design encryption algorithm based on chaos and the self-synchronizing method of stream cipher. The study result will be a reference for digital chaotic cipher application and provide a feasible scheme for digital video encryption transmission and monitoring system.

Keywords: chaos, video encryption transmission, chaotic synchronization, self-synchronizing stream cipher

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得黑龙江大学或其他教育机构的学位或证书而使用过的材料。

学位论文作者签名：董学洋　　签字日期：2010 年 6 月 9 日

学位论文版权使用授权书

本人完全了解黑龙江大学有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权黑龙江大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

学位论文作者签名：董学洋　　导师签名：

签字日期：2010年 6 月 9 日　　签字日期：2010 年 6 月 9 日

学位论文作者毕业后去向：深圳华为技术有限公司

工作单位：深圳华为技术有限公司　电话：13043653466

通讯地址：深圳市龙岗区坂田华为基地　邮编：518129

第1章 绪 论

1.1 课题研究背景及意义

21 世纪是一个信息时代，伴随信息化的不断深入与扩展，人们对信息安全性的要求也在日益增长。因此密码学成为人们不断关注的热点研究方向。在信息化建设中，针对于特殊应用需求，由于网络的透明性以及其他不安全因素，我们往往希望能够将我们要传送的数据，图像以及视频等信息进行有效的加密传输，从而避免重要信息的泄漏。视频传输以及视频监控也是信息化建设中一个主要的领域。视频传输与一般数据文件传输一样，可以直接采取加密手段进行遮掩，防止在网络中被截获。对于传统视频监控系统来说，其传输的图像信息以模拟信号为主，由于对图像的处理和传输均采用模拟技术，因此不但图象质量低，而且严重浪费系统资源，难以组成复杂的网络结构，监控功能可扩展性较差。因此传统的模拟集中监控方式已经不能满足人们的需求，随着监控规模的不断扩大，监控系统的视频图像显示效果与视频监控要求的实时性之间的矛盾也越来越显著。随着计算机软硬件技术以及视频压缩编码技术的快速发展，以计算机为基础的数字视频监控系统逐渐走进我们的生活[1]。由于要实现数字视频信息的加密传输与监控，实时的视频信号数据量过大，对现有的数据存储能力和信号传输能力提出了极大的挑战。为此，需要对视频信号进行压缩，通过减少原始视频信息中的冗余信息，如时间、空间、视觉等来使用更少的信息比特来表达原始视频信息，在研究过程中不断地提出了各种视频压缩算法。因此选择一个较好的视频压缩算法是传输加密视频信号的必要前提。另一方面，随着社会对信息需求面的不断增多，对视频信息安全性需求也变的越来越急迫。由于视频信息具有数据量大，易于进行交流等特点，因此它也最容易被理解。一些信息对个人而言有隐私的需要，对商业公司的各种信息产品而言有版权保护的需要，对国家要害部门

而言又有国家安全方面的需要。因此它给我们带来方便的同时，也给我们带来了隐患，敏感信息可能轻易地被窃取、篡改、非法复制和传播等。这些需要都和视频信息的加密是分不开的。因此视频信息安全问题已成为人们关心的焦点，视频信息安全保护技术的研究也显得尤为重要。

传统密码学中的密码算法主要包括：对称密钥以及非对称密钥加密。对称密码算法又可分为分组密码和流密码[2]。流密码算法比较简单，并且易于实现，适用于要就加密速度快的实时性传输中。流加密算法中的密钥流发生器是信息安全领域中的重要研究课题，其核心是可以产生“随机”密钥流的发生器。混沌是由确定系统产生的貌似随机的动力学行为[3]。混沌密码学是研究基于混沌系统产生的伪随机密钥流进行信息加密与解密的新兴学科，混沌系统的参数和对信息加密变换中使用的参数构成了密钥集，以达到抗攻击的目的。混沌轨道对初始条件和系统参数的敏感性，使其生成的密钥流满足传统密码设计系统的扩散与混淆原则，混沌轨道的长期不可预测性和混沌密码编码的便捷性使其满足传统密码设计的安全性与有效性的原则，混沌密码学为信息安全保障提供了新的工具[4]。精心设计的混沌系统可生成高复杂度的密钥序列，与视频明文数据进行加密后可获得难以破译的密文。这样在网络中传输的视频文件就能够抵抗截获者的分析。

对于实时的视频信号传输与监控则要求更高一些，既要实现数字信号的压缩与编码，又要完成加密与解密运算，最重要的是如何实现加解密视频监控信号的实时同步，保证视频信号具备加密特性的同时能够在远程不间断的进行实时监控。

我们的研究课题是基于国家自然科学基金项目“基于 FPGA 技术的混沌序列密码加密芯片的研究（60672011）”基础上进一步将该网络加密卡应用于实际中来，旨在设计出一个基于混沌算法的视频文件等数据加密传输系统，并设计实现一种实时视频流加密的视频监控系统。该系统是一种工作于宽带网

络环境中的视频压缩及加密传输系统，它既可以用于普通 Internet 的环境中，用于对普通用户的隐私信息进行保护；还可以用于党政机要部门专用的网络环境中，防止敏感信息被窃取；也可用于大型企业集团内部网络，如电信行业、银行系统，防止商业机密的泄露。另外本课题还将就混沌算法的混沌同步通信技术相关理论进行探讨研究，课题的研究将进一步丰富混沌保密通信的理论，希望可以加快混沌保密系统的实用化进程，使其能够开发成为一种实际产品，应用于信息安全与保密通信中，更好的服务于国家和社会，同时也为相关领域的研究工作者提供参考。

1.2 国内外同类课题研究现状及发展趋势

当前在网络数据传输中的密码算法主要有公钥密码算法和私钥密码算法两大类：私钥密码算法分为分组密码（Block cipher）和序列密码（Stream cipher）算法[5]。国际标准化组织（International Standard Organization，ISO）将分组密码中的 DES（Data Encryption Standard）算法定为私钥数据加密的标准[6]。DES 的加密速度很快，可以通过硬件电路实现，而且适合大数据量的加密。但通过分析，分组密码中相同的明文组总是对应于相同的密文组，所以密码分析者可以利用明文冗余度进行解密，存在安全性问题。RSA（Rivest Shamir Adlemen）算法，是最早发明的公钥密码系统之一，已被 ISO 推荐为公钥数据加密标准[7]。它的安全性依赖于大素数因式分解的困难，但是其协议的实现以及运算时间都很长，不利于用硬件来实现。当前的基于以上传统密码算法的加密技术大都是利用软件来实现，比如先将文件进行加密再存储到磁盘上，当需要读取时再利用软件应用程序进行解密；还有的是采取专用密码芯片，将密码算法集成到芯片中。但这两种方法事实上都是软件加密，在加密速度上都受到计算机 CPU 工作频率的限制，因此并不适用于大数据流量的视频流加密。为了解决软件加密对实时加密速度方面的限制，国内外许多公司也都先后提出硬件专用加密芯片，比如美国 3com 公司的 3CR990-T-95 加密网卡

芯片，Intel 公司的 IntelPRO/100S 系列网卡以及国内北大青鸟的 SJY01A 系列网卡[8]。这些加密网卡尽管实现了硬件加密，但由于采用的是专用密码算法芯片，首先从密码算法的更新角度看不利于算法移植，一旦需要算法升级则需重新开发设计；另外其采用的密码算法大都是国际通用算法，对于需要安全级别较高的部门来说不够可靠。

在应用领域，网络是流媒体传输的主要途径，如何保证压缩视频流的安全性成为流媒体的一个重要课题。目前基于视频压缩算法的视频传输系统相继问世，技术也日趋成熟，但大多数此类产品仅实现视频信息的压缩及传输功能，面向普通视频数据的应用，不具有保密功能，无法防止视频信息被窃取、攻击。而对视频数据的加解密涉及到密码学的应用。出于历史和政治的原因，对密码的公开研究历史并不长，因而极大地阻碍了视频信息安全性研究的发展。近年来，随着互联网在线交易的广泛开展[9]，以及对视频信息保密需求的增多，这种局面正在逐渐改变，对视频信息安全系统的研究也越来越成为视频信息领域里的一个重要课题。

视频加密算法在上个世纪 70 年代被提出，在 90 年代后期开始被广泛研究。就其发展过程来看，可以分为三个阶段[10]。

第一个阶段主要是研究置乱算法，即直接置乱图像或视频数据，以达到使数据混乱而不能被理解的目的。但这些置乱过程改变了图像象素间的统计关系，从而不利于压缩处理，因此仅适合不需要压缩编码的应用中。

第二个阶段，利用通用密码算法对压缩的视频数据加密传输。随着多媒体技术的发展，多媒体压缩编码标准不断提出，如 JPEG 图像压缩标准、MPEG、H.264 等视频压缩标准。因此可以将视频数据先经过压缩编码，再使用 DES、IDEA、RSA 等密码算法来实现加密，最后进行传输。但由于这些算法复杂，不易硬件实现，将会带来不可忽视的运算负荷和时延，因此不能满足多媒体实时传输播放的要求。

第三个阶段，采用部分加密方法来加密视频数据，即选择视频码流中的一部分数据进行加密。选择加密少量数据可以降低计算复杂度，来满足实时性要求。然而这种加密方法却存在安全隐患：对图像编码数据，只对I帧加密，P帧和B帧不加密，由于P帧和B帧中会有帧内预测的宏块并且由于帧间的相关性，P帧、B帧中有未加密的I块。因此此类加密方法也被认为是不安全的。

序列密码当明文相同时可以产生不同的密文，明文的统计特性将被扰乱，不存在扩展明文序列和错误传递等方面缺点。序列密码满足一次一密的加密特点，而且易于硬件实现，已成为当前密码系统中研究与应用的热点。近年来，由于混沌序列具有容易产生，对初值敏感以及类似于白噪声等特点，被认为是序列密码的很好选择。目前国内外对于混沌算法应用于数据加密传输系统中的研究已经开展，但是从总体上来说目前基于混沌的加密算法的实现都是采用基于软件的方式完成的，从加密速度和实时性上来说并不理想。我们设计的基于混沌算法的网络数据加密系统则是利用 FPGA 硬件技术实现快速流加密，利用混沌序列对数据进行加密传输，从硬件实现以及加解密速度上都有较好表现。针对实时视频流数据的加密传输以及监控，我们整个系统无论是视频压缩编码还是加密实现均采用硬件技术实现，而且通过 FPGA 密码芯片的可编程、易于算法移植的优势，我们利用自同步流密码方案将其实现，从加密速度和实时性要求的角度来说都有着重要的意义。

数字网络化将会是新时代视频传输的发展趋势，面向未来更加广阔的信息化时代，针对敏感用户需求的视频加密技术也将必然是视频监控领域未来崭新发展的新方向。除了传统高强度分组密码(如 DES、IDEA、RSA 等）外，基于混沌序列密码及流密码算法的研究也将成为视频加密传输系统中的新领域和新方向。

1.3 本文主要研究内容

论文的主要内容包括：

1、密码学基础以及混沌理论研究

从理论上研究密码学相关概念，传统密码学分类，包括分组密码和序列密码。着重分析了序列密码理论，探讨了几种流密码设计方案。混沌算法由于具备长期不可预测的伪随机序列特性，因此近年来被广泛应用到密码设计研究中，从混沌理论的定义、特性和判别角度分析了混沌算法理论，为我们的加密算法设计提供理论保证。

2、网络数据加密卡中混沌加密算法的设计

采用基于 Matlab/Simulink 的方案设计我们的混沌加密算法，实现网络数据加密卡的加密核心。将结合我们的系统需求设计一种基于 logistic 算法的混沌序列密码产生方法，应用到我们设计的改进型 A5/1 算法中并将其下载到 FPGA 网卡芯片中，从而最终实现网络数据加密卡的硬件加密方法。

3、实时视频数据加密传输的自同步方案研究与设计

因为我们采用的是同步方式的流加密算法，即收发端始终保持同步，发端加密数据不断在收端进行解密，只要通信双方的密钥序列产生器具有相同的种子密钥和相同的初始状态，就能产生相同的密钥序列，从而实现准确解密。但是在视频流加密传输系统的设计中，能否实现随时随意对加密的视频数据进行实时解密是我们需要解决的一个重要方面和难点所在，因为我们要求接收端能够随时启动来解密当前时刻发送端的加密数据，而不是始终与发送端保持同步。考虑到此类情况的解决，我们研究了流加密算法中的自同步流密码方式，设计一种自同步方案，能够使通信双方的密钥序列产生器自动地恢复同步，从而能够正确解密。

4、视频加密传输系统构建

1) 系统硬件框图设计

视频流数据的采集使用如下的硬件接口：硬件是基于三星公司(Samsung)的摄像头和杭州海康威视有限公司的音/视频硬压缩卡，视频编码是采用了最新的视频编码标准 H.264 来实现的，通过将摄像头采集来的原始 YUV 视频流数据进行 H.264 编码，来进一步的压缩图像，以便于能够利用网络进行传输；经压缩编码后的视频流数据再经过网络数据加密卡进行加密操作，经 Internet 传输至远程监控客户端的视频数据服务器，由服务器端的网络数据解密卡进行解密运算，再将解密后的视频数据流存储，以供客户端调用监控。

2) 系统软件接口设计

视频采集部分软件接口是音/视频压缩采集卡开发商提供的硬件驱动程序(软件开发包 SDK)。通过使用此软件开发包 SDK 来辅助编写视频采集系统的视频采集端和客户端软件。

5、本论文就混沌算法其它相关理论进行探讨，研究了混沌同步保密通信技术，对混沌遮掩以及混沌键控等技术进行了研究和仿真。

1.4 本文的结构安排

第 1 章为绪论。主要对视频加密课题研究背景与意义、国内外相关领域研究状况及文章主要研究内容和结构进行了阐述。

第 2 章为密码学基础与混沌理论研究。首先详细介绍了序列密码的加密原理以及分类方式，分析了当前同步流密码的密钥序列生成器的几种方案。然后给出了混沌的定义，总结了混沌的特性，归纳了混沌方程的判别方法；最后分别从几种判别方法上对 logistic 以及 Lorenz 混沌方程进行了仿真实验。

第 3 章为混沌同步保密通信研究。探讨了混沌同步保密通信，对混沌遮掩以及混沌键控保密通信的几种方法进行了仿真实现。

第 4 章为混沌序列密码算法及自同步技术的研究实现。详细介绍了我们设计的 logistic 离散混沌序列密码原理以及实现方案。对于要求实时性较高的加密传输系统，根据自同步流密码的原理，我们又用硬件实现了一种自同步

流密码方案，并通过仿真验证了同步结果。

第 5 章为视频加密传输系统的研究与实现。详细介绍了我们利用的 SDK 开发包中的 API 函数以及各部分设计过程，最后实现了针对板卡的 SDK 二次开发以及 H.264 格式视频文件的视频解码播放器的二次开发。最后经过测试，设计实现了视频加密传输系统的服务器端与客户端软件。

第2章 密码学基础与混沌理论

密码学的发展主要经历了三个阶段：古代密码方法、古典密码和近代密码[11]。计算机和现代数学方法一方面为加密技术提供了新的概念和工具，另一方面也给破译者提供了有力武器。计算机和电子学时代的到来给密码设计者带来了前所未有的自由，脱离了最初的手工阶段，并且不用受到机械式和电子器件时代密码设计的限制，利用电子计算机可以设计出更为复杂的密码系统[12]。

20 世纪 70 年代以来,现代密码学伴随着下面两个众所周知的里程碑式研究逐渐发展起来：作为一种商业加密标准 DES 的提出与建立，公钥加密方案的提出与 RSA 加密方法。现代密码系统是一种跨多个学科的研究领域，涵盖了数学知识，信息理论，通信技术和网络技术等。

2.1 密码学基本概念

密码学（cryptology）是研究信息系统安全保密的科学，包括密码编码学和密码分析学。研究如何对信息进行编码，实现对信息的隐蔽的技术叫做密码编码学（cryptography）。密码分析学（cryptanalysis）是主要研究加密消息的破译或消息的伪造，不知道密钥的情况下从密文推出明文或密钥，从而破译密文的技术。

在密码学中，有五个基本组成要素：明文、密文、密钥、加密算法和解密算法[2]。

明文(Plaintext)：信源输入的原始信息，通常用 m 或 p 表示。明文空间用 P 表示。用某种方法伪装消息以隐藏明文的内容的过程称为加密(Encryption)，被加密的消息称为密文(Cipher text)，而把密文转变为明文的过程称为解密(Decryption)。

密文(Cipher text)：是明文加密输出结果，通常用 c 表示。密文空间用 C

来表示。

密钥（key）：加密或解密所需要的除密码算法之外的关键信息,是参与密码变换的参数 k。密钥空间用 K 表示。

加密算法(Encryption Algorithm)：密码员对明文进行加密操作时所采用的一组规则称作加密算法，也称编码过程（用 E 表示，$c=E_k(p)$）。

解密算法(Decryption Algorithm)：接收者对密文解密所采用的一组规则，也称解码过程（用 D 表示，$p=D_k(c)$）。

密码系统（cryptosystem）：是加密与解密组成的系统，就是发方明文初始值与密钥作为加密变换的输入，经过一定的加密处理以后得到密文。对于一个密码系统而言，总是要求它满足：对于任意 $k\in K$，有加密算法 $e_k:P\to C$ 和相应的解密算法 $d_k:P\to C$，使得 $e_k\in E$ 和 $d_k\in D$ 分别为加密与解密函数，满足 $d_k(e_k(x))=x$，这里 $x\in P$。即用加密算法得到的密文总是能用相应的解密算法恢复出原始的明文来。而密文消息的获取同时依赖于初始值和密钥。一个完整的密码系统由密码算法、信源、信宿和攻击者组成[2]。加密系统模型如图2-1所示。

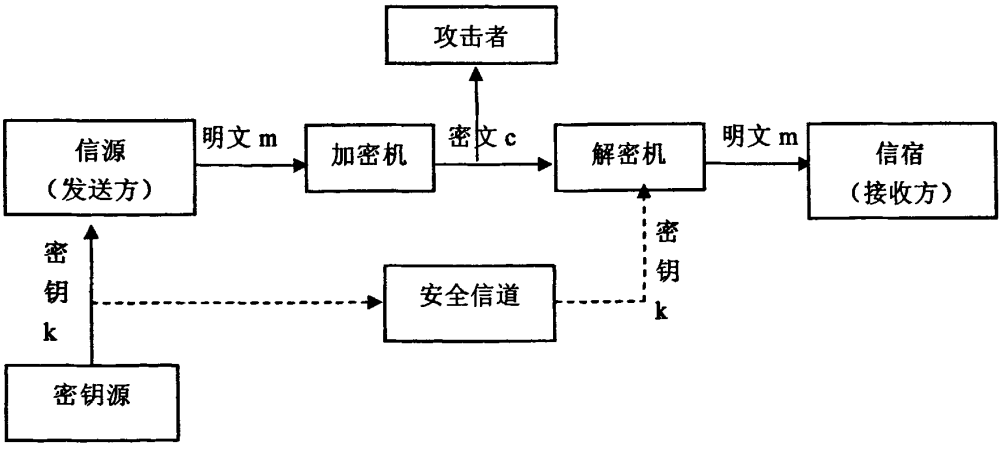


图2-1 加密系统框图

Figure2-1 The block diagram of encryption system

2.2 密码算法分类

密码算法常见有三种分类方式[11]。

1．明文变换到密文的操作类型

此类加密算法主要有两种基本操作。

（1）替代（substitution）：将明文中的元素映射成为其他元素，是一种非线性变换。

（2）换位/置换（transposition）：重新排列明文中的元素，是一种线性变换，置换操作是可逆的。

2．对称和非对称加密

（1）对称加密（Symmetric Encryption）：也称为秘密密钥密码、单密钥密码或传统密码。即加密密钥与解密密钥相同，或实质上等同，即从一个易于推出另一个。

（2）非对称加密（Asymmetric Encryption）：也称为公开密钥密码、双密钥密码。即发送者和接收者各自使用一个不同的密钥，加密密钥与解密密钥不同，形成一个密钥对，用其中一个密钥加密的结果，可以用另一个密钥来解密。公钥密码体制的发展是整个密码学发展史上最伟大的一次革命，它与以前的密码体制完全不同。这是因为：公钥密码算法基于数学问题求解的困难性，而不再是基于代替和换位方法；另外，公钥密码体制是非对称的，它使用两个独立的密钥，一个可以公开，称为公钥（public key），另一个不能公开，必须为密钥持有人秘密保管,称为私钥（private key）。

3．分组密码和序列密码

对称密钥密码又可分为分组密码和序列密码。

（1）分组密码（block cipher）：也称块加密，一次处理一块（组）元素的输入，对每个输入块产生一个输出块，即一个明文分组被当做一个整体来产生一个等长的密文分组输出。通常使用的是 64 位或 128 位的分组大小。

常见分组加密算法有 DES、IDEA、RC6、AES 等。

（2）序列密码（stream cipher）：也称为流密码，即连续地处理输入元素，并随着该过程的进行，一次产生一个元素的输出，即一次加密一个比特或一个字节。常见流密码算法有 A5、E0、RC4、Helix、SEAL、WAKE 等。

与分组密码相比，序列密码具有如下优点：（1）容易硬件实现，加解密速度快；（2）错误扩散低，更适用于准确度要求高的传输环境中；（3）适用于接收端需要缓冲或单个字符处理应用中(如远程传输)。

2.3 序列密码理论

2.3.1 序列密码原理

序列密码即流密码。流密码通信框图如图 2-2 所示[13]。

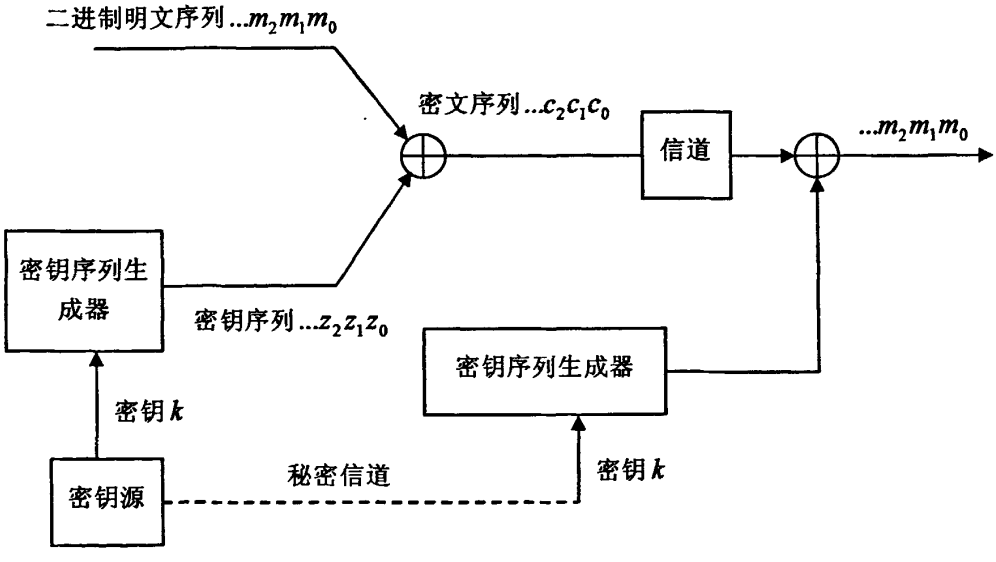


图 2-2 流密码通信框图

Figure2-2 The block diagram of stream cipher communication

在流密码中，将明文 m 写成连续的符号，利用密钥流中的第 i 个元素 k_i 对应明文中的第 i 个元素 m_i 进行加密，若加密变换为 E，则加密后的密文为：

$$c = E_k(m) = E_{k_1}(m_1)E_{k_2}(m_2)\cdots E_{k_i}(m_{k_i})\cdots \qquad (2\text{-}1)$$

设与加密变换 E 对应的解密变换为 D，其中 D 满足：

$$D_{k_i}(E_{k_i}(m_i)) = m_i, \qquad i = 1,2,\cdots \qquad (2\text{-}2)$$

则通过解密运算可译得明文为：

$$m = D_k(c) = D_{k_1}(E_{k_1}(m_1))D_{k_2}(E_{k_2}(m_2))\cdots = m_1 m_2 \cdots \qquad (2\text{-}3)$$

密文 C 可由明文 m 与密钥 k 进行模 2 加获得。因此要用该密码系统通信就要求每发送一条消息都要产生一个新的密钥并在一个安全的信道上传送，通常称这种通信系统为“一次一密系统”。

序列密码与分组密码的区别就在于有无记忆性。序列密码的滚动密钥 $z_0 = f(k,\sigma_0)$ 由函数 f、密钥 k 和指定的初态 σ_0 完全确定。此后，由于输入加密器的明文可能影响加密器中内部记忆元件的存储状态，因而 $\sigma_i (i>0)$ 可能依赖于 $k,\sigma_0,x_0,x_1,\ldots,x_{i-1}$ 等参数。

2.3.2 序列密码分类

根据内部记忆元件的存储状态 σ_i 是否依赖于输入的明文字符，流密码可进一步分成同步和自同步方式两种。σ_i 独立于明文字符的叫做同步流密码，否则叫做自同步流密码[14]。

1、同步流密码

在同步流密码中，密钥序列是独立于消息序列而产生的，其输出与明文无关。其加解密过程可用图 2-3 表示。

加密过程可描述为：$s_{i+1} = f(s_i,k)$，$z_i = z(s_i,k)$，$c_i = h(z_i,m_i)$。这里 s_0 是初始状态，k 是密钥，f 是 s_i 到 s_{i+1} 的状态转移函数，z 是产生密钥序列 z_i 的函

数，h是将明文m_i加密成密文c_i的加密函数。

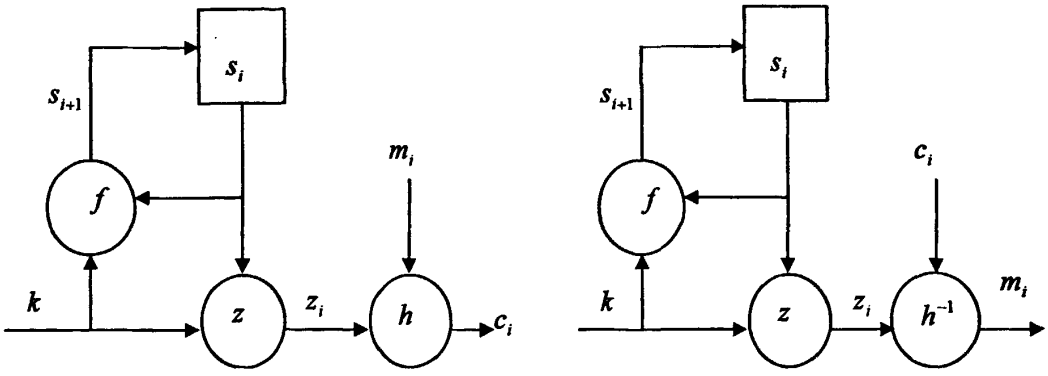


图 2-3 同步流密码模型

Figure2-3 The model of synchronization stream cipher

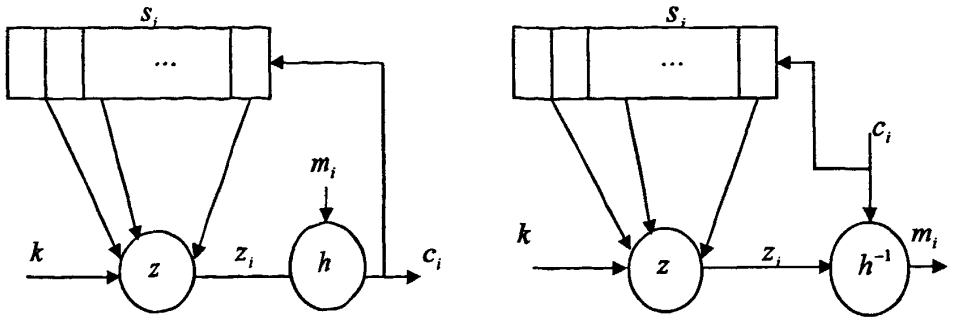
对应上述加密过程，可以得到同步流密码的解密过程描述为：$s_{i+1}=f(s_i,k)$，$z_i=z(s_i,k)$，$m_i=h^{-1}(z_i,c_i)$。这里h^{-1}表示加密函数h对应的解密函数。

同步流密码方案中，若发方和接收方的初始密钥以及内部状态相同，则将产生同样的密钥序列。同步流密码的一个优点是没有错误传播，在传输密文的信道中，噪声或其他的原因带来的一个密文比特的错误传输仅仅影响该密文比特所对应的一个明文比特位，因此不会影响从后面的密文正确解密。在有限域$GF(2)$上开展的模二加法流密码是目前最广泛使用的流密码体制，加密函数可表示为$c_i=z_i\oplus m_i$。

2、自同步流密码

密钥序列与生成的有限数量的密文有关的流密码方案被称为自同步流密码[15-16]。与同步流密码不同，自同步流密码具有错误传播有限的特性，在密文传输过程中，修改、插入、删除一个密文比特，将影响后续有限数目的密文正确解密。因为自同步流密码系统采用密文反馈的方式，因此该系统密码

分析工作十分复杂，不易进行。自同步流密码的加解密过程可用图 2-4 表示。



(a) 加密　　(b) 解密

图 2-4 自同步流密码模型

Figure2-4 The model of self-synchronizing stream cipher

自同步流密码的加密过程可描述为：$s_{i+1}=(c_{i-t},c_{i-t-1},\cdots,c_{i-1})$，$z_i=z(s_i,k)$，$c_i=h(z_i,m_i)$。自同步流密码中，内部反馈状态的每一位输入序列都是前面 n 位密文的函数。该密码方案的输出函数决定其复杂性，内部反馈移位寄存器控制输出函数 z 产生伪随机序列。自同步流密码在信道传输中若出现一比特出错，它将在反馈移位寄存器中移动 n 位，影响 n 位输出密钥，解密时会影响 n 位明文序列。在自同步方案中，接收端如果正确连续地接收了 n 位密文，在相同初始密钥作用下就能够产生相同的密钥序列，实现自同步解密。这种自恢复同步性使得它对攻击者的一些主动攻击不像同步流密码那样敏感。但它将明文每个字符扩散在密文多个字符中而强化了其抗统计分析的能力。

在同步流密码中，由于 $z_i=f(k,\sigma_i)$ 与明文字符无关，因而此时密文字符 $y_i=E(z_i,x_i)$ 也不依赖于此前的明文字符。因此，可将同步流密码的产生分成密钥流生成和加密变换两个部分。

我们实际讨论的数字保密通信系统一般都是二元系统，因而在有限域 GF(2)上讨论的二元加法序列密码是目前最为常用的序列密码体制，其加密变

换为异或运算，表示为 $y_i = z_i \oplus x_i$。

实际密码设计中，密码设计者的最大愿望是设计出一个良好的密钥生成器，使得密钥经其扩展生成的密钥流序列具有以下几个性质：极大的周期性、良好的统计特性、抵抗线性密码分析和统计分析。

密钥流生成器的设计目的是用一个短的随机密钥(也称种子密钥) k 生成一个长的密钥流，用这个长的密钥流对明文加密或对密文解密，从而使一个短的密钥可用来加密更长的明文或解密更长的密文的目的。

2.3.3 密钥流生成器

将密钥流生成器分成驱动部分和非线性组合部分（如图 2-5 所示）[2]。

驱动部分控制生成器的状态转移，并为非线性组合部分提供统计性能好的序列；而非线性组合部分要利用这些序列组合出满足要求的密钥流序列。

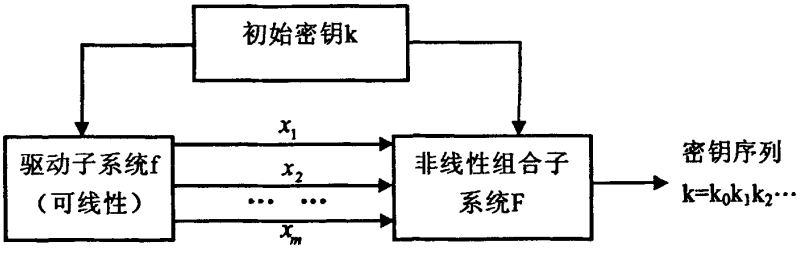


图 2-5 密钥流生成器原理图

Figure2-5 The schematic diagram of key stream generator

目前最为主要的密钥流产生器如图 2-6 所示，其驱动部分是一个或多个线性反馈移位寄存器。

移位寄存器是流密码产生密钥流的一个主要组成部分。GF(2)上一个 n 级反馈移位寄存器由 n 个二元存储器与一个反馈函数 $f(a_1,a_2,\ldots,a_n)$ 组成。

每一存储器称为移位寄存器的一级，在任一时刻，这些级的内容构成该反馈移位寄存器的状态，每一状态对应于 GF(2)上的一个 n 维向量，共有 2^n 种可能的状态。

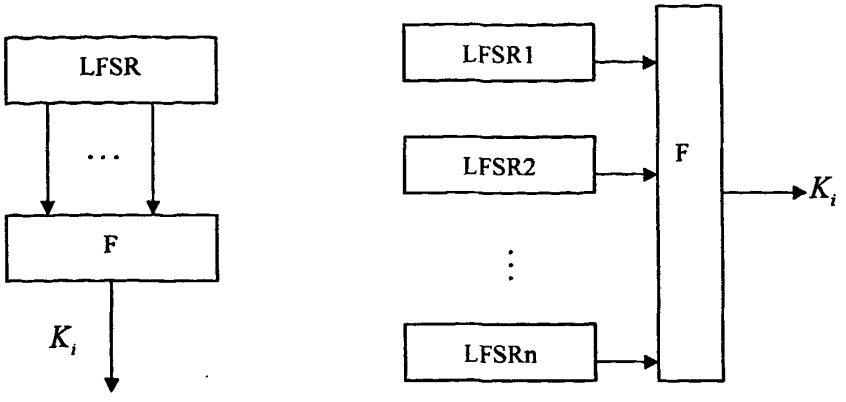


图 2-6 常见密钥流生成器

Figure2-6 Common key stream generators

如果移位寄存器的反馈函数 $f(a_1,a_2,\ldots,a_n)$ 是 $a_1,a_2,\ldots,a_n$ 的线性函数，则称之为线性反馈移位寄存器 LFSR（linear feedback shift register）[17]。此时 f 可写为 $f(a_1,a_2,\ldots,a_n)=c_n a_1 \oplus c_{n-1}a_2 \oplus \cdots \oplus c_1 a_n$，其中常数 $c_i=0$ 或 1，$\oplus$ 是模 2 加法。$c_i=0$ 或 1 可用开关的断开和闭合来实现。线性反馈移位寄存器因其实现简单、速度快、有较为成熟的理论等优点而成为构造密钥流生成器的最重要的部件之一。

线性反馈移位寄存器输出序列的性质完全由其反馈函数决定。n 级线性反馈移位寄存器最多有 2^n 个不同的状态。若其初始状态为 0，则其状态恒为 0。若其初始状态非 0，则其后继状态不会为 0。因此 n 级线性反馈移位寄存器的状态周期小于等于 2^{n-1}。其输出序列的周期与状态周期相等，也小于等于 2^{n-1}。只要选择合适的反馈函数便可使序列的周期达到最大值 2^{n-1}，周期达到最大值的序列称为 m 序列。

流密码的安全性取决于密钥流的安全性，要求密钥流序列有好的随机性，因此密码分析者就无法对它预测。也就是说，即使截获其中一段，也无法推测后面是什么。但因为密钥流是周期的，要完全做到随机性是困难的。严格地说，这样的序列不可能做到随机，只能要求截获比周期短的一段时不会泄

露更多信息，这样的序列称为伪随机序列。

Golomb 对伪随机周期序列提出其应满足的以下 3 个随机性假设[18]：

① 在序列的一个周期内，0 与 1 的个数相差最多是 1。

② 在序列的一个周期内，长度为 i 的游程占游程总数的$1/2^i$(i=1,2,…)，且 0 和 1 的任一相同游程个数最多差 1。

③自相关函数是一个常数。

假设①说明$\{a_i\}$中 0 与 1 出现的概率整体上相同；②说明 0 与 1 在序列中每一位置上具有相同概率；③意味着通过对序列与其平移后的序列做比较，不能给出其他任何有用信息。

m 序列满足 Golomb 的 3 个随机性公设。

①在一个周期内，0、1 出现的次数分别为$2^{n-1}-1$和2^n-1。

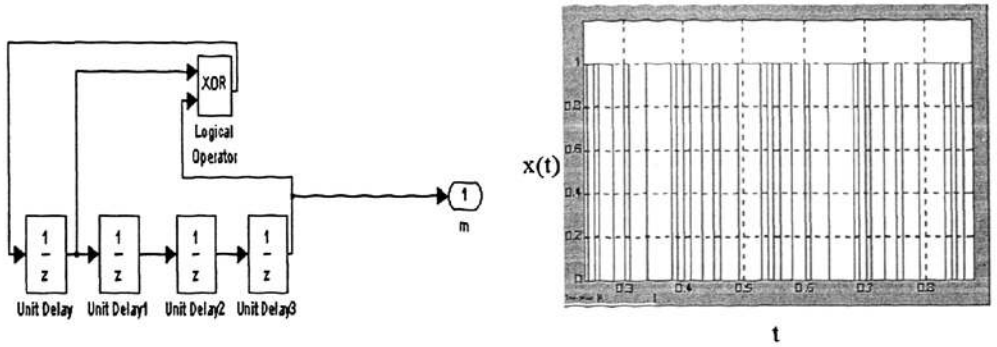
②在一个周期内，总游程数为2^n-1；对$1\le i\le n-2$，长为 i 的游程有$2^{n-i}-1$个，且 0、1 游程各半；长为 n-1 的 0 游程一个，长为 n 的 1 游程一个。

③$\{a_i\}$的自相关函数为

$$R(\tau)=\begin{cases}1, & \tau=0\\ -\dfrac{1}{2^n-1}, & 0<\tau\le 2^n-2\end{cases} \qquad (2\text{-}4)$$

图 2-7 是利用 matlab 的 simulink 设计的一个产生 4 级 m 序列的仿真图以及波形图。本仿真设计的 m 序列级数为 4，即其周期为$2^4-1=15$。初始状态：$(a_3,a_2,a_1,a_0)=(1,1,0,0)$。循环输出序列为：000111101011001。

尽管 m 序列满足 Golomb 三条随机性公设，但通常不能直接利用它作为一个流密码的密钥序列。因为对 m 序列来说，知道其少量的序列位以后就可以预测后面的数列了。例如比较著名的解法：Berlekamp-Massey 迭代算法，简称 B-M 算法[14]。在此不详细叙述了。



(a) m 序列仿真图　　(b) 波形图

(a) the simulation of m sequence　　(b) the waveform

图 2-7 4 级 m 序列仿真图及波形图

Figure2-7 The simulation and waveform diagram of 4-level m sequence

前面已经介绍过流密码密钥生成器可分解为驱动子系统和非线性组合子系统，驱动子系统常用一个或多个 LFSR 来实现，非线性组合子系统用非线性组合函数 F 来实现。

常见的非线性序列密码生成器有以下几种：

1、Geffe 序列密码生成器

Geffe 序列密码生成器[14]如图 2-8 所示：由 3 个 LFSR 组成，其中 LFSR2 为控制生成器。

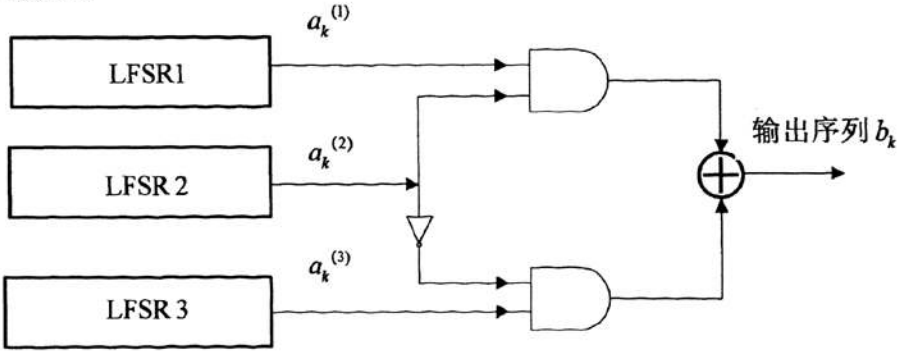


图 2-8 Geffe 序列生成器

Figure2-8 Geffe sequence generator

当 LFSR2 输出 1 时，LFSR2 与 LFSR1 相连接；当 LFSR2 输出 0 时，LFSR2

与 LFSR3 相连接。若设 LFSRi 的输出序列为 $\{a^{(i)}_k\}$ $(i=1,2,3)$，则输出序列 $\{b_k\}$ 可以表示为

$$b_k = a_k^{(1)}a_k^{(2)} + a_k^{(3)}\overline{a}_k^{(2)} = a_k^{(1)}a_k^{(2)} + a_k^{(3)}a_k^{(2)} + a_k^{(3)} \qquad (2\text{-}5)$$

Geffe 序列的周期实现了极大化，且 0 与 1 之间的分布大体上是平衡的。

2、J-K 序列密码触发器

J-K 触发器[14]如图 2-9 所示，它的两个输入端分别用 J 和 K 表示，其输出 c_k 不仅依赖于输入，还依赖于前一个输出位 c_{k-1}，即

$$c_k = \overline{(x_1 + x_2)} \cdot c_{k-1} + x_1 \qquad (2\text{-}6)$$

其中 x_1 和 x_2 分别是 J 和 K 端的输入。J-K 触发器的真值表见表 2-1。

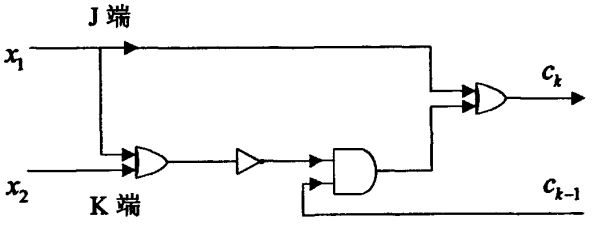


图 2-9 J-K 触发器

Figure2-9 J-K Flip-flop generator

表 2-1 J-K 真值表

Table 2-1 Table of J-K true value

J	K	c_k
0	0	c_{k-1}
0	1	0
1	0	1
1	1	$\overline{c}_{k-1}$

利用 J-K 触发器设计的非线性序列密码生成器见图 2-10。

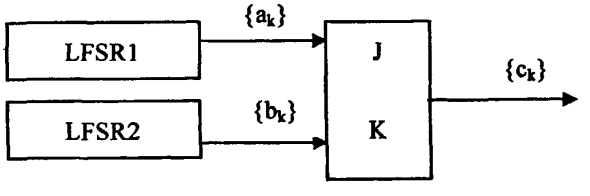


图 2-10 J-K 序列生成器

Figure2-10 J-K sequence generator

令驱动序列 $\{a_k\}$ 和 $\{b_k\}$ 分别为 m 级和 n 级 m 序列，则有

$$c_k = \overline{(a_k + b_k)} \cdot c_{k-1} + a_k = (a_k + b_k + 1) \cdot c_{k-1} + a_k \qquad (2\text{-}7)$$

如果令 $c_{-1}=0$，则输出序列的最初 3 项为

$$\begin{aligned} c_0 &= a_0 \\ c_1 &= (a_1+b_1+1)a_0+a_1 \\ c_2 &= (a_2+b_2+1)((a_1+b_1+1)a_0+a_1)+a_2 \end{aligned} \tag{2-8}$$

当 m 与 n 互素且 $a_0+b_0=1$ 时，序列 $\{c_k\}$ 的周期为 $(2^m-1)(2^n-1)$。

3、钟控序列生成器

钟控序列最基本的模型是用一个 LFSR 控制另外一个 LFSR 的移位时钟脉冲，如图 2-11 所示[14]。

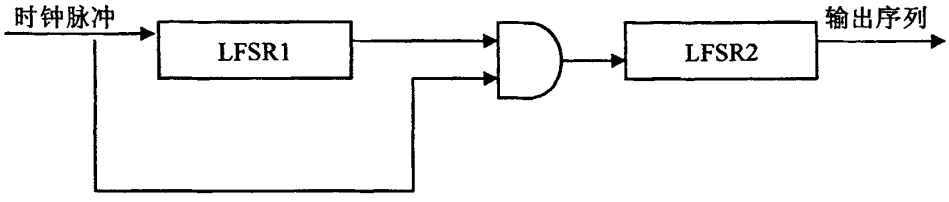


图 2-11 钟控序列生成器

Figure2-11 clock control sequence generator

实际应用中，可以用上述最基本的钟控序列生成器构造复杂的模型。

从以上的几个序列密码模型来看，要想设计一个性能良好的序列密码并不是十分容易的。其中最基本的设计理念是“密钥流生成器的不可预测性”，主要包括下述几种基本原则：

① 长周期以及较高线性复杂度。

② 具有良好的统计性能。

③ 满足密码学的“混乱”与“扩散”特性。

④ 能够抵抗不同形式的密码分析攻击。

2.4 混沌学基础理论

1963 年美国气象学家爱德华·诺顿·洛伦茨提出蝴蝶效应：南美洲一只

蝴蝶扇一扇翅膀，就会在佛罗里达引起一场飓风。这就是混沌理论（Chaos theory）中一个著名表述[19-20-21]。混沌理论作为自然科学的一个分支，是在数学和物理学中，研究非线性系统在一定条件下表现出的“混沌”现象的理论。

在非线性科学中，“混沌”这个词的含义和本意相似但又不完全一致，非线性科学中的混沌现象指的是一种确定的但不可预测的运动状态。它的外在表现和纯粹的随机运动很相似，即都不可预测。但和随机运动不同的是，混沌运动在动力学上是确定的，它的不可预测性是来源于运动的不稳定性。或者说混沌系统对无限小的初值变动和微扰也具有敏感性，无论多小的扰动在长时间以后，也会使系统彻底偏离原来的演化方向。混沌理论认为在混沌系统中，初始条件十分微小的变化，经过不断放大，对其未来状态会造成极其巨大的差别。

2.4.1 混沌的定义

混沌理论是非线性科学的核心部分，它的理论及应用价值具有很广泛的研究价值，但是迄今为止，对混沌概念还没有公认的严格的定义，通常对混沌概念的界定都是从混沌现象的本质特征入手，即从数学和物理两个层次上考察，通过比较非周期、有界和对初值的敏感性来定义混沌系统。 比较常用的有以下几种定义方法：

定义 1：Li-Yorke 混沌定义

1975 年，华裔数学家李天岩(Tianyan-Li)和他的导师约克(Yorke)在《美国数学月刊》中发表了一篇论文，题目是《Period Three Implies Chaos》—《周期 3 意味着混沌》[22]，用数学的方法定义了“混沌（Chaos)”，并且第一次使用了 Chaos 这个词，是数学界比较公认的定义方法。混沌的李-约克定义如下：设有连续自映射 $f:I\to I$，I 是 R 中的一个闭区间，如果满足 f 的周期点的周期无上界；存在不可数集合 $S\in I$，满足

（1）任给 $x_1,x_2\in S,x_1\neq x_2$ 时,有 $\limsup_{x\to\infty}\left|f^n(x_1)-f^n(x_2)\right|>0$

（2）任给 $x_1, x_2 \in S$ 有 $\liminf_{x\to\infty} \left| f^n(x_1) - f^n(x_2) \right| = 0$

（3）任给 $x_1 \in S$ 及 f 的任意周期点 x_2，有 $\limsup_{x\to\infty} \left| f^n(x_1) - f^n(x_2) \right| > 0$

这里 n 表示 n 重函数运算。称 f 在 S 上是混沌的[23]。由李-约克的定义可见，他们是用非周期、有界和对初值的敏感性这三个方面的本质特征来对混沌进行刻画的。

定义 2：Devaney 混沌定义

R.L.Devaney 采用拓扑方法给出了混沌的定义[3]。

（1）拓扑传递性。$f: R^n \to R^n$ 称为是拓扑传递的，如对任何一对开集 $U, V \subset R^n$，存在 $k > 0$，使得 $f^k(U) \cap V \neq \Phi$。其中，$f^k(U)$ 表示迭代 k 次的函数。

（2）对初始条件的敏感依赖性。$f: R^n \to R^n$ 有对初始条件的敏感依赖性，如存在 $\delta > 0$，对任何 $x \in R^n$ 和 x 的任何领域 N，存在 $y \in N$ 和自然数 n，使得 $\left| f^n(x) - f^n(y) \right| > \delta$。

（3）周期点在 V 中稠密。

定义 3：连续流混沌

对于系统 $\dot{x} = F(x)$，以 x_0 为初始值的解表示为 $\varphi(t, x_0), \varphi(0, x) = x$。记 $\Gamma(1) = \varphi(1, x), \Gamma(2) = \varphi(2, x), \cdots, \Gamma(n) = \varphi(n, x), n \in N$，如果 $\Gamma(1), \Gamma(2), \cdots, \Gamma(n), \cdots$ 是混沌序列，则系统 $\dot{x} = F(x)$ 是混沌的[24]。

但是，在实际研究混沌运动规律，证明某一系统是否会有混沌现象产生时，我们往往不是通过混沌的数学定义来确定，而是通过混沌的一些性质去研究和判断。以下将介绍一下混沌的几个主要特征。

2.4.2 混沌的性质

混沌理论作为一个科学理论,它具有三个关键概念，或者说是三个特性：

初值敏感性、分形（fractals）和奇异吸引子[25]。

1、初值敏感性（蝴蝶效应）

混沌现象表征了现实世界不可琢磨的复杂性，从而对科学决定论进行了反驳。混沌理论发现，对于一些系统，只要初始条件稍有偏差或进行微小的扰动，系统的最终状态会出现巨大的差异。系统的输出状态为混沌时，即使初始值输入仅有非常微小的差别，其输出状态经过有限时间迭代后仍将会以指数速率迅速分离。因此不可预测混沌系统的长期演化行为。这种对初始值的敏感特性即被称为蝴蝶效应。

2、分形（fractals）

分形是由著名数学家 Mandelbrot 在分形几何理论中提出的概念。它的含义是系统在不同尺度下具有一种自相似的性质。自相似性意味着递归特性，即在一个模式内部还有一个相同模式，可产生出具有特定结构和规则的隐含的有序模式。分形具有二个主要特征：第一，自始至终都是不规则的；第二，在不同的尺度上，不规则程度却是一个常量。

3、奇异吸引子

吸引子是系统被吸引并最终固定于某一状态的一种形态。有三种不同的吸引子控制和限制物体的运动程度：点吸引子、极限环吸引子和奇异吸引子（即混沌吸引子或洛伦兹吸引子）。点吸引子与极限环吸引子都起着限制的作用，使系统产生静态的、平衡的特征，故也称收敛性吸引子。奇异吸引子使系统偏离收敛吸引子的区域，诱发不同形态。它具有复杂的拉伸、折迭与伸缩的结构，可以使指数型发散保持在有限的空间中；它使系统变为非预设模式，从而使系统成为不可预测性的。

2.4.3 混沌的判定

根据混沌的几个特性，目前对混沌的判定主要有以下几种途径：

（1）奇异吸引子与相平面图：动力学系统的状态通常用一组都与时间由

关的变量 x,y,z 等来描述，以状态变量为坐标形成的空间称为相空间。根据 Matlab 的 ODE 程序进行仿真试验或用 Runge-Kutta(龙格库塔)法对系统进行数值仿真试验，可以得到系统的奇异吸引子和相平面图。根据相点的运动轨迹或变化情况，可以凭经验对系统的运动特性进行初步判断。但该方法只是对混沌系统的定性估计。

（2）功率谱分析：依据随机信号处理理论，包含所有可能的频谱成分的是随机运动，而一切具有一定的频谱结构的为非随机运动。按照功率表达式绘制系统的功率谱图，如果频谱图具有一个或几个尖峰，则系统为周期或拟周期系统；如果没有明显的峰值或尖峰连在一起，则系统是混沌的[26]。

（3）Lyapunov 指数：混沌系统最基本的特征之一是运动轨迹对初始条件的敏感依赖性。而李雅普诺夫(Lyapunov)指数[3]可以表征系统运动的特征，它沿某一方向取值的大小和正负表示相邻轨道沿该方向平均收敛或平均发散的快慢程度。对于高维的动力学系统，有多个 Lyapunov 指数，称为 Lyapunov 指数谱。其中，每个指数对应于某一特定方向上的轨道收敛率或发散率。通常来讲，Lyapunov 指数的个数等于相空间的维数。对于多维混沌系统，它的 Lyapunov 指数谱中应至少有一个为正。因此，可以通过计算 Lyapunov 指数来定量判断混沌系统特性。

（4）分岔图：倍周期分岔理论是研究非线性常微分系统由于参数的改变而引起解的不稳定性，从而导致解的数目变化的方法。如果某个动力系统结构是不稳定的，则任意小的、适当的扰动都会使系统的拓扑结构发生巨大的变化，称这种质的变化为分岔。用这种思想可以临摹系统在整个演化过程中所表现出来的动态画面。

（5）Kolmogorov 熵（K 熵）

测度熵[23]是由柯尔莫戈洛夫（Kolmogorov）于 1958 年提出来的，也称 K 熵，是用于对混沌系统混沌特性的识别及度量的方法。由于混沌轨道存在的

局部不稳定性，其在相空间中按指数速率迅速分离，轨道的初始点十分靠近，因此在开始时不能通过测量来区分，但它们完全分离后就能够区分了。通过K熵的分析，混沌系统带来的信息量与可区分的相异混沌轨道数目N有关。对于混沌系统来讲，N与时间成指数正比变化，表示为$N \propto e^{Kt}$，K为测度熵，表征混沌信息发生的速率。混沌序列向量组的维数决定了系统测度熵的大小，向量组的维数越多，满足分辨率条件的向量在总向量中的比重越小，此时系统的测度熵也越小，混沌系统的随机性越强，被恢复的可能性越小，复杂度也就越高。但增加了系统的复杂度，带来了较大的计算量。若向量组的维数较少，满足分辨率条件的向量在总向量中的比重越大，此时系统的复杂度较小，混沌系统的随机性越弱，被恢复的可能性较大，系统的计算量较小。以下将介绍几种常见混沌映射，并将对其特性进行仿真分析。

2.5 常见混沌系统示例

1.logistic 混沌映射[3]

（1）周期倍增分叉现象：

对于函数

$$X_n = \mu X_{n-1}(1-X_{n-1}) \ , \mu \in [0,4], x \in [0,1], 其中 n = 1,2,3... \qquad (2\text{-}9)$$

其中，随着参数μ的增大，先是只有周期1的稳定解；当μ增大到μ_1时，周期1的稳定解分叉为2个周期2的稳定解；当μ增大到μ_2时，2个周期2的稳定解又分叉为4个周期4的稳定解……当μ增大到μ_m时，周期2^{m-1}的稳定解又分叉为2^m个周期2^m的稳定解……如此继续。这就是倍周期分岔进入的混沌，是一种典型的非平衡过程产生的混沌。

解释倍周期分岔现象，我们从混沌描述中最重要的一维非线性迭代方程入手。这类方程中最有典型意义的是logistic方程，即方程（2-9），也成虫口

方程。式中的μ是与虫口增长率有关的控制参数，同时它的大小也反映了系统非线性的强弱。

周期倍增过程没有限制，可以一直这样分下去，但对应的μ值却有一个极限μ_∞,到达μ_∞时，迭代的稳定解是2^∞周期解—周期无穷大，即没有周期。所以这时得到的是非周期解，迭代的数据将会随机分布在整个区间内，无法把握，系统进入混沌状态。下图所示即为 Logistic 映射$f(x)=4\cdot x(1-x)$，初始值为 0.1 时以x_i为横轴坐标，$f(x)$为纵坐标，迭代次数为 1000 次绘制的轨迹分布图和迭代过程如图 2-12。

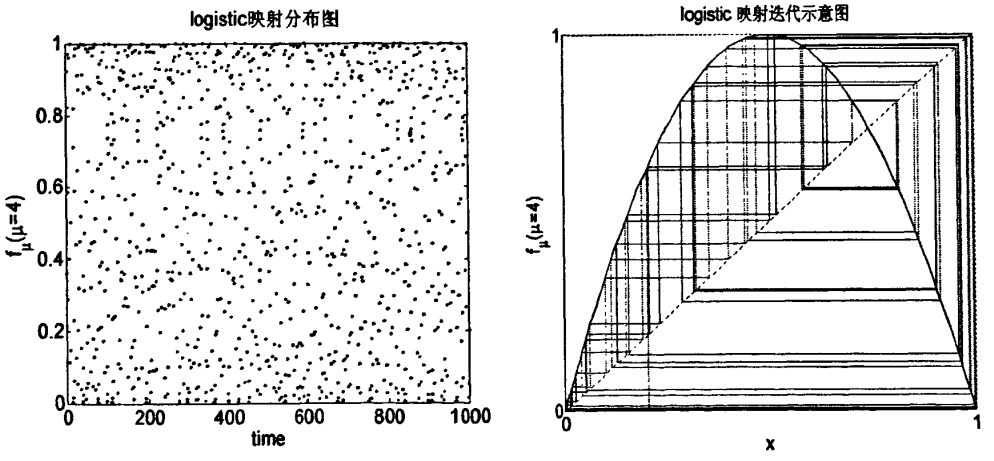


图 2-12 logistic 映射轨迹图及迭代图

Figure2-12 The track and iteration figure of logistic map

在研究混沌理论中，Feigenbaum 常数也是一个重要发现[27]。美国康奈尔大学的物理学家菲根鲍姆（Feigenbaum），发现了在周期倍增分叉现象中更深层次的规律，从而揭示出系统从有秩序转向混沌的秘密。菲根鲍姆发现：在周期倍增分叉过程中，随着分叉次数 M 的增加，相邻的两个分叉点μ_m和μ_{m+1}的间距$\Delta m=\mu_{m+1}-\mu_m$组成一个渐进的等比数列，分叉宽度ξ_m也组成一个渐进的等比数列，并且这两个等比数列都有极限。菲根鲍姆测出了这两个等比数列的公比，它们的倒数分别叫做菲根鲍姆常数δ和菲根鲍姆常数α，它们分别是：

$\delta = 4.669201.....$，$\alpha = 2.50290.....$。他曾说过，周期倍增分叉现象和规律的发现，大大地改变了人类对宇宙的认识。

（2）一维映射的 Lyapunov 指数计算

混沌系统初值敏感性是指相空间中初始距离很近的两条轨迹会以指数速率发散，Lyapunov 指数即是根据相轨迹有无扩散运动特征来判别系统的混沌特性。

对于一维映射，初值 x_0 和它的邻近值 $x_0+\delta x$，经 n 次迭代后两点之间的距离以指数速度分离，Lyapunov 指数用来度量这种分离性，如式(2-10)所示：

$$LE = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|f'(x_i)\right| \tag{2-10}$$

Lyapunov 指表征的是运动沿迭代轨道平均收敛或平均发散的结果，其数值可以大于、小于或等于零，一维系统只有一个 Lyapunov 指数值为正表明在每个局部运动轨道都是不稳定的，对初值极其敏感，相邻轨道将会迅速以指数分离。因此，$LE>0$ 表明系统是混沌的；$LE=0$ 系统是稳定的，属于定周期运动；$LE<0$ 表明系统轨道有周期，并且对初始条件不敏感。LE 由负到正，表明系统运动转变为混沌[28]。

图 2-13 所示为 logistic 映射的分岔图以及 lyapunov 指数图。通过对比我们可以清晰看到 logistic 映射随着参数 μ 的变化由周期运动逐步走向混沌运动状态。

从图中可以看出开始是周期加倍分岔(也称周期倍化分岔或周期倍分岔)，然后是混沌，混沌区中又有周期窗口。窗口放大后又可见到同样结构的一套东西。此就是混沌理论中的无穷自相似结构。

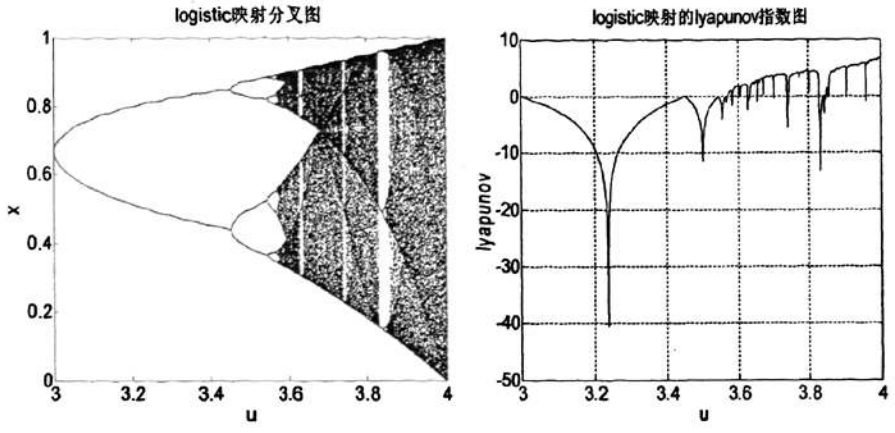


图 2-13 Logistic 分岔图以及 lyapunov 指数图

Figure2-13 The bifurcation and lyapunov exponent figure of logistic map

当$1<\mu\le 3$时，不管初值是多少，经过足够长的迭代，结果都会达到同一个确定值$x_\infty \to 1-1/\mu$。这个值就叫做周期一或不动点，对应的$LE\le 0$。

继续增大参数μ，当$\mu>3$，周期一点不再稳定。初值稍有变化，迭代的结果就再也不会回到周期一点$1-1/\mu$，而出现了周期二，LE<0。

让参数μ再增大，当$\mu=3.449$时，周期二解也变的不稳定了，取而代之的是稳定的周期四解。当参数继续增大，使得$\mu=3.544$时，周期四解又变的不稳定了，取而代之的是稳定的周期八解。一直迭代下去，还会出现周期十六、周期三十二等等。

当 3 .56994< u ≤ 4 时，logistic 映射由周期解转向混沌运动态。但从分岔图以及 LE 图谱上不难发现，在 3 .56994< u ≤ 4 区间内还存在着非混沌区，因此经过对 LE 图μ取值细化编程分段测验，证得 LE>0，即 logistic 映射处于混沌区时μ值的取值范围有如下区间：(3.570－3.582)、(3.584－3.605)、(3.607－3.626)、(3.635－3.655)、(3.674－3.701)、(3.703－3.738)、(3.744－3.828)、(3.850－3.854)、(3.857－3.905)、(3.907－3.960)、(3.962－4.000]等。

（3）Kolmogorov 熵计算

算法流程如下：

对于一个长度为 N 的序列样本空间$[u(1),u(2),\cdots,u(N)]$，定义 m 维向量组

$x(1),x(2),\cdots,x(i),\cdots,x(N-m+1)\in R^m$ 其中 $x(i)=[u(i),u(i+1),\cdots,u(i+m-1)],1\le i\le N-m+1$

定义以下几个中间量：

m 维向量 x(i)和 x(j)的最大距离

$$d[x(i),x(j)]=\max_{k=1,2,\cdots,m}\left(\left|u(i+k-1)-u(j+k-1)\right|\right) \tag{2-11}$$

满足与第 i 个 m 维向量 x(i)的最大距离小于 r 的向量数

$$C_i^m(r)=\frac{(\text{满足}d[x(i),x(j)]\le r\text{的}j\text{的个数})}{\mathrm{N}-m+1} \tag{2-12}$$

根据$C_i^m(r)$,定义$\Phi^m(r)$，$\Phi^m(r)=\dfrac{1}{N-m+1}\sum_{i=1}^{N-m+1}\ln C_i^m(r)$ (2-13)

则测度熵可以被定义为 $K(\mathrm{m},r,N)=\Phi^m(r)-\Phi^{m+1}(r)$ (2-14)

那么下面以 Logistic 方程为例来说明一维分辨率 r 对 logistic-Kolmogorov 的关系变化曲线的影响。

r 是用来描述区分混沌轨道程度的参数，当 r 较小的时候，说明此时的分辨率较高，即能够将距离较近的轨道区分出来，但此时系统的复杂度较高，混沌系统的测度熵，即混沌系统所产生的信息量是巨大的,系统需要较大的计算量。当 r 较大的时候，此时的分辨率较低，只能将相距很远的混沌轨道才能区分开来，此时系统的复杂度较低，混沌系统的测度熵较小。下图是 logistic 混沌系统的 Kolmogorov 熵相对于分辨率 r 的变化关系曲线。在这里我们设定测度熵测定系统的输入序列空间样本为 logistic 方程产生的数据，分辨率的初始值为 0.2，步进值为 0.001，终止值为 0.9，向量组维数为 2。Logistic-Kolmogorov 相对于 r 的关系变化曲线如图 2-14 所示。

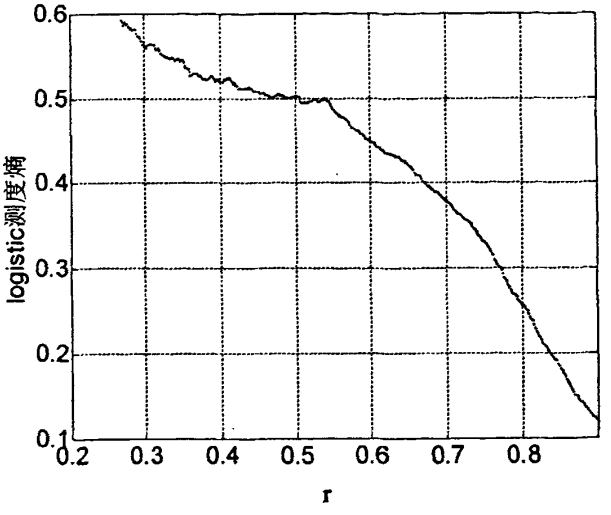


图 2-14 Logistic 映射的 Kolmogorov 熵图

Figure2-14 The Kolmogorov entropy figure of logistic map

从 logistic 混沌映射的 Kolmogorov 熵变化图可以看出，随着 r 的增大，分辨率降低，混沌系统的测度熵随之减小，系统的复杂度及系统产生的信息量降低。

2.Lorenz 混沌映射

著名的 Lorenz 系统[17]是如下方程组：

$$\begin{cases} \dot{x} = -ax + ay \\ \dot{y} = cx - y - xz \\ \dot{z} = xy - bz \end{cases} \tag{2-15}$$

其中，a ,b ,c 是常量参数，x ,y ,z 是自变量 t 的函数，$\dot{x},\dot{y},\dot{z}$ 分别表示 x,y,z 关于 t 的导函数。系统参数取值为 $a=10,b=8/3,c=28$；x 的初始值为 1.2，y 的初始值为 1.3，z 的初始值为 1.6。

（1）Lorenz 吸引子的相平面图以及平面投影图

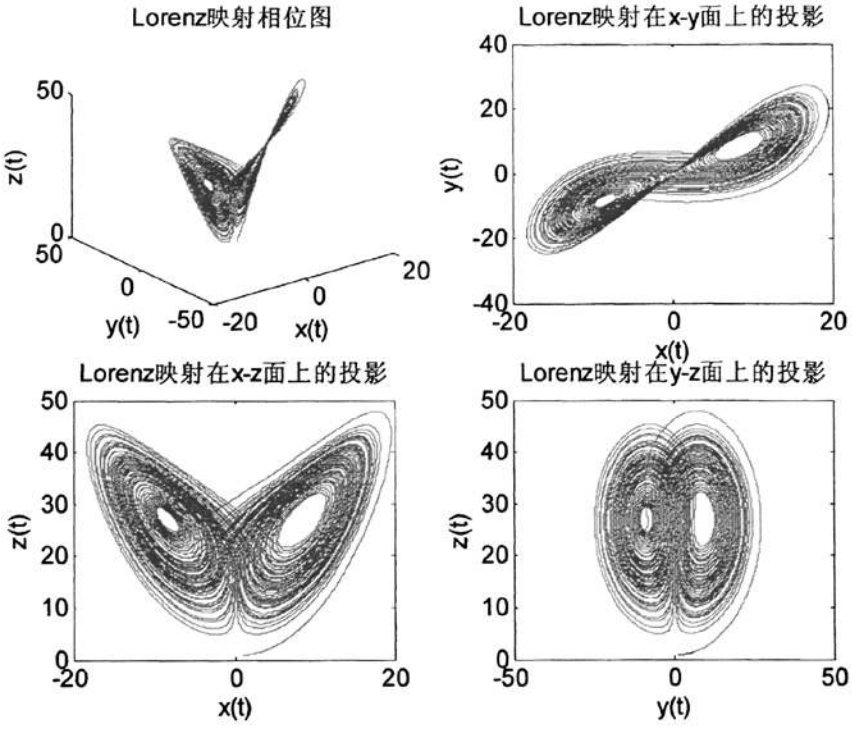


图 2-15 Lorenz 映射相位及投影图

Figure2-15 The phase and projection figure of Lorenz map

在 Matlab6.5 下利用 ODE 程序进行仿真试验，得到 Lorenz 吸引子的三维相平面图以及在 x-y ,x-z ,y-z 平面上的投影图如图 2-15 所示。

（2）多维映射的 Lyapunov 指数计算

三维混沌方程如式(2-16)

$$\begin{cases} x_{n+1} = f_1(x_n, y_n, z_n) \\ y_{n+1} = f_2(x_n, y_n, z_n) \\ z_{n+1} = f_3(x_n, y_n, z_n) \end{cases} \tag{2-16}$$

若初始点 (x_0, y_0, z_0) 的偏差为 $(\delta x_0, \delta y_0, \delta z_0)$,从初始点出发，利用上述公式依次迭代得到 (x_1, y_1, z_1) (x_2, y_2, z_2).....(x_n, y_n, z_n)。

其偏导数构成的雅克比矩阵特征值分别为 $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{n-1}$，所以高维映射的

n 个 Lyapunov 指数分别为：

$$LE_1 = \frac{1}{n}\ln|\lambda_1| \quad LE_2 = \frac{1}{n}\ln|\lambda_2| \cdots LE_n = \frac{1}{n}\ln|\lambda_n| \tag{2-17}$$

计算得到高维映射的 Lyapunov 指数

$$LE = \frac{1}{n}\ln(\frac{\|\Delta x_n\|}{\|\Delta x_0\|}) = \frac{1}{n}\ln(\lambda_{n-1}, \lambda_{n-2}, \cdots \lambda_0) = \frac{1}{n}\sum_{i=0}^{n-1}\ln\lambda_i \tag{2-18}$$

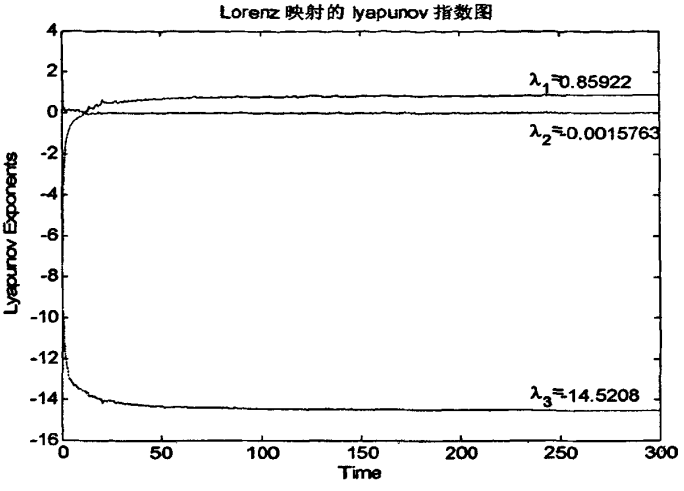


图 2-16 Lorenz 映射的 lyapunov 指数图

Figure2-16 The lyapunov exponent figure of Lorenz map

图 2-16 为 Lorenz 映射的 lyapunov 指数图谱，其中，系统参数取值为 $a=10, b=8/3, c=28$。计算得出 Lorenz 映射的 lyapunov值分别为 $\lambda_1 = 0.85922$、$\lambda_2 = -0.0015763$、$\lambda_3 = -14.5208$。

对于多维混沌映射来说，若产生混沌现象，则最大的 Lyapunov 指数为正。只要有一个 Lyapunov 指数值为正的，就可判断为混沌系统，如果有两个或以上 Lyapunov 指数为正的，则称其为超混沌系统，超混沌系统具有的强大保密特性使其在在保密通信领域具备更广泛的应用空间[29]。

（3）Lorenz 映射的 Kolmogorov 熵计算

图 2-17 是 lorenz 混沌系统的 Kolmogorov 熵相对于分辨率 r 的变化关系曲

线。在这里我们设定测度熵测定系统的输入序列空间样本为 Lorenz 方程产生的数据，分辨率的初始值为 3，步进值为 0.1，终止值为 10。

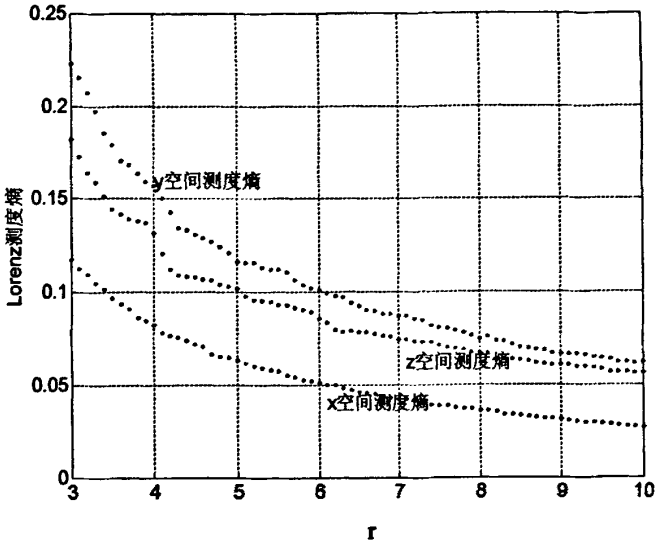


图 2-17 Lorenz-Kolmogorov 相对于 r 的变化曲线

Figure2-17 The variation curve of Lorenz Kolmogorov relative to resolution

其中横轴坐标为 r，纵轴坐标为 Lorenz 测度熵，图中的三条曲线分别代表 xyz 空间的测度熵。从图中我们可以看出，随着 r 的提高，x，y，z 空间测度熵都有下降趋势。即随着系统分辨率的降低，系统的复杂度及系统产生的信息量随之下降。

2.6 本章小结

本章介绍了密码学的相关理论以及混沌算法理论。传统密码算法包括对称密码和非对称密码，其中对称密码学又包括分组密码和序列密码，序列密码相对于分组密码来说，具备以下优点：易于硬件实现，加解密速度快；错误扩散低，适宜于高准确传输环境以及接收端需要缓冲的或单个字符处理的应用中(如远程传输)。然后详细介绍了序列密码的加密原理以及分类方式，包含同步以及自同步两种。同步流密码的密钥流生成器直接决定着密钥序列的

特性，根据其设计的主要方法，分析了当前同步流密码的密钥序列生成器的几种方案。混沌理论作为非线性科学的一个分支，因其伪随机特性以及对初始条件高度敏感越来越受到密码学家的广泛关注。本章首先给出了混沌的几种定义方法，总结了混沌的特性，归纳了混沌方程的判别方法；最后分别从几种判别方法上对 logistic 以及 Lorenz 混沌方程进行了仿真实验，从而在理论上和数值分析角度深入的了解了混沌理论。

第3章 混沌同步保密通信研究

混沌理论在保密通信中有一个广泛的研究领域，那就是混沌同步保密通信。混沌同步技术一般是在模拟电路中进行研究与实现的。尽管目前这项新技术的研究尚处于实验室研究阶段，但由于它的实时性强，保密度高，运算速度快等明显优势，已经显示出其在保密通信领域中的强大生命力，混沌学家已经注意到混沌在同步保密通信中的应用价值。

自从Pecaro 和 Carroll用实验证明了两个耦合混沌系统可以实现同步[30]，大量研究人员已经对混沌应用于通信领域进行了日益广泛的研究。 利用混沌载波在较高带宽上传输数字信号，其结果继承了扩频通信的很多优势，例如减缓多径衰落、降低错误检测率。混沌保密通信方式主要有三种：一、非相干的混沌保密通信，该方法利用混沌直接完成保密通信；二、相干的混沌保密通信，该方法利用同步的混沌系统实现；三、混沌数字编码通信。第二种混沌同步方案为目前主要进行研究的领域，主要包括有：混沌遮掩（Chaos Masking）[31]、混沌调制（Chaos Modulation）[32]、混沌移位键控（Chaos Shift Keying）[33]和混沌扩频通信（Chaos Spread Spectrum Communication）[34]技术，这几种方案都基于混沌同步实现的。其中混沌遮掩将明文信息与混沌伪噪声信号叠加，实现对有用信号的隐藏，属于模拟通信；其余三种为数字通信。混沌移位键控方案中，混沌开关键控（COOK, Chaos On-Off Keying）利用开关关闭来表征二进制信号信息；CSK 利用混沌信号来代表二进制信息进行传输；差分混沌键控（DCSK, Differential Chaos Shift Keying）利用混沌参考信号间相位变化代表信号信息进行传输。

3.1 混沌遮掩同步保密通信

混沌遮掩通信[35-36]传送模拟信号的思想是发送方以混沌同步为基础，把小信号叠加在混沌信号上，利用了混沌信号的伪随机特性，把信息信号较好

地隐藏在看似杂乱无章的混沌信号中，并在接收端用一个与发送方同步的混沌信号解调出信号信息，以此实现保密通信。根据采用的掩盖方式不同，分为相加 $s(t)=x(t)+c(t)$ 、相乘 $s(t)=x(t)*c(t)$ 和加乘结合 $s(t)=(1+kx(t))*c(t)$。

接收方利用接收到的信号来驱动本地混沌系统，使其与发送端混沌系统达到同步，通过同步后的混沌信号与接收到的遮掩信号进行反变换就能够得到原始信息，完成保密传输。

优点：实际中的研究表明，混沌掩盖保密通信系统电路实现简单，同步是具有鲁棒性的，同步对驱动信号的小扰动不敏感。传输信号的幅值较小时，保密性较好。

缺点：这种混沌掩藏通信方法要求混沌载波的功率要远大于信号功率，当有用信号幅度较大时，混沌信号的波形上就将泄漏出原始信息的特征，失去了保密功能，因此只能传输小能量信号。容易受到外界噪声的影响。

3.2 混沌键控同步保密通信

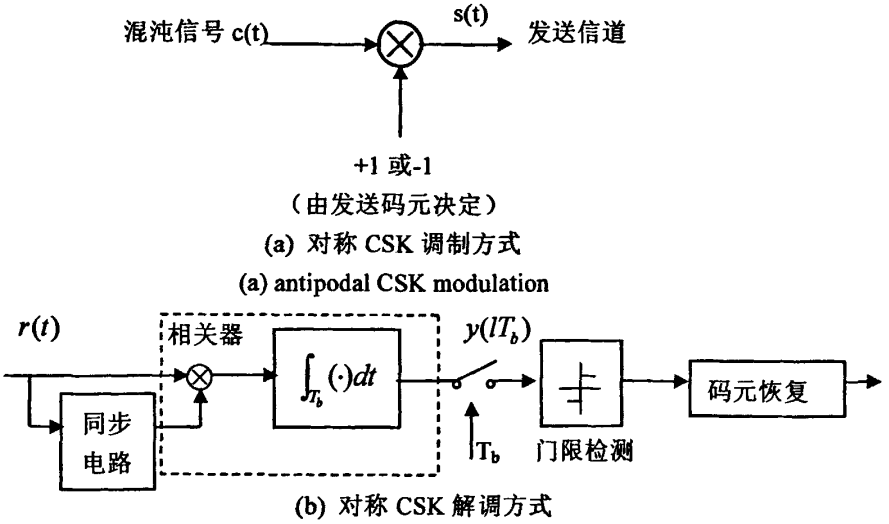
与模拟调制相比较，混沌数字调制技术更加实用，因此先后有大量的基于混沌的数字调制和解调技术被研究。通常来讲，在混沌数字通信系统中，数字码元被映射到非周期的混沌函数中。例如在混沌移位键控(CSK：chaos shift keying)中[37]，不同的码元符号映射到不同的混沌吸引子上，他们或是由一个不同分叉参数的动态系统产生，或是由一组完全不同的动态系统产生。CSK 的解调方式有两种，分别是相干解调和非相干解调。由于相干解调技术在接收端重构的混沌信号要求与发送端的混沌信号精确一致，但在实际应用中，严格的鲁棒性的同步技术还不能实现，因此相干解调技术不能够在实际环境中来实现。非相干解调技术的实现则不需要在接收端重构混沌信号，因此在实际应用中更加灵活。CSK 调制机制中的非相干解调技术是由 Hasler and

Schimming 在 2000 年提出的。这种方式是基于通过选择码元符号降低后验概率，使误码率最佳的最优分组来实现的。基于此类解调机制的还有一种差分混沌键控（DCSK：differential chaos-shift-keying）的键控通信方式，它是由 Kolumbán 等人 1996 年提出的[38]。在 DCSK 方式中，发送端发送一个参考混沌信号，再紧随发送一个由二进制码元调制的相同的信号。在接收端，这两段混沌信号是相关的，因此根据相关器输出的符号判断来对发送二进制码元进行解调。

3.2.1 混沌移位键控（CSK）调制

数字二进制码元传输系统的 CSK 调制方式可以用一个或两个混沌发生器来完成。

（一）采用一个混沌发生器的 CSK 调制方式



(a) 对称 CSK 调制方式

(a) antipodal CSK modulation

(b) 对称 CSK 解调方式

(b) antipodal CSK demodulation

图 3-1 对称 CSK 的调制与解调

Figure3-1 The modulation and demodulation of antipodal CSK

CSK 调制分为两种：对称 CSK（ACSK：antipodal CSK）方式[39]及混沌开关键控 COOK（chaotic on-off-keying）方式。

图 3-1 描述了相关的对称 CSK 的调制与解调方式。在调制端，首先产生混沌信号 $c(t)$，T_b 是一个位持续时间段，在第 l 个符号持续时间 $[(l-1)T_b, T_b]$ 内，如果传输的是“+1”，则发送混沌信号；如果传输的是“-1”，则发送一个与原混沌信号反相的信号。因此，发送端信号可以表示为：

$$s(t)=\begin{cases}c(t) & \text{发送的符号位为+1}\\ -c(t) & \text{发送的符号位为-1}\end{cases} \tag{3-1}$$

假设传输信号受到噪声干扰，接收到的信号由下式给出，其中 $\xi'(t)$ 代表的是噪声信号：

$$r(t)=s(t)+\xi'(t) \tag{3-2}$$

在接收端，通过一个自同步电路来实现混沌信号的重构，重构信号与接收到的传输信号 $r(t)$ 具有相关性。相关器的输出为下式：

$$y(lT_b)=\int_{(l-1)T_b+T_s}^{lT_b} r(t)c(t)dt \tag{3-3}$$

T_s 是实现同步的获取时间。相关器的输出和阈值（这里为 0）进行比较来确定接收到的信号是“+1”还是“-1”。如果相关器的输出值大于 0，检测值为“+1”；否则为“-1”。

混沌开关键控 COOK[40]调制方式原理图如图 3-2 所示。COOK 的调制器由一个开关控制。如果传输的是符号“+1”，则开关关闭，传送的是混沌信号 $c(t)$；如果是符号“-1”，则开关打开，不传输任何信号。因此发送信号可表示为：

$$s(t)=\begin{cases}c(t) & \text{发送的符号位为+1}\\ 0 & \text{发送的符号位为-1}\end{cases} \tag{3-4}$$

接收端解调可以采用相干或非相干解调方式。相干解调的方法与对称 CSK 解调采用的是一样的方式。非相干解调的原理如下图 3-3 所示。

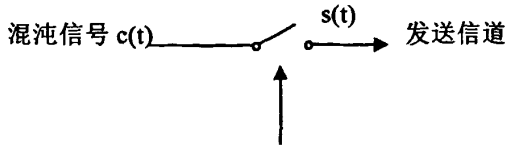


图 3-2 COOK 调制端

Figure3-2 The modultion of COOK

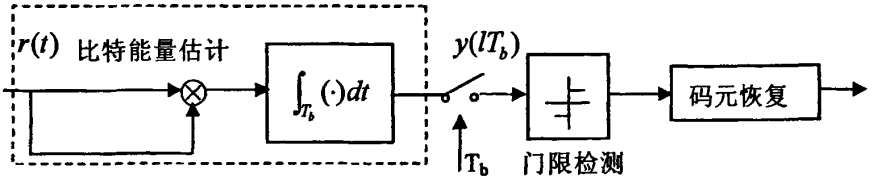


图 3-3 基于位能量检测的解调器

Figure3-3 The demodultion based on bit emergy detect

非相干解调器事实上就是一个位能量估值器，其第l个符号位的输出由下式确定：

$$y(lT_b)=\int_{(l-1)T_b}^{lT_b} r^2(t)dt=\int_{(l-1)T_b}^{lT_b} s^2(t)dt+2\int_{(l-1)T_b}^{lT_b} s(t)\xi'(t)dt+\int_{(l-1)T_b}^{lT_b}[\xi'(t)]^2dt \qquad (3\text{-}5)$$

然后输出值传输至阈值检测器，阈值设置为 0 和$E[\int_{(l-1)T_b}^{lT_b} c^2(t)dt]$的中值，也就是$E[\int_{(l-1)T_b}^{lT_b} c^2(t)dt]/2$，其中的$E[]$指的是数学期望的计算。如果估算位的能量大于这个阈值，则解码为“+1”；否则为“-1”。但当有噪声加入的时候，阈值就需要提升以实现最佳性能。

（二）两个混沌发生器构成的 CSK 调制方式。

如图 3-4，根据传输符号“+1”和“-1”来相应的传送混沌信号$c(t)$和$s(t)$，用方程（3-6）表示[41-42]：

$$s(t)=\begin{cases} c(t) & \text{发送的符号位为+1} \\ s(t) & \text{发送的符号位为-1} \end{cases} \qquad (3\text{-}6)$$

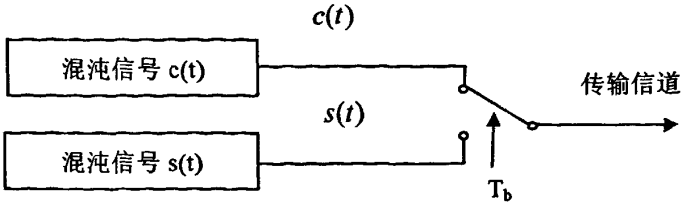


图 3-4 双混沌信号 CSK 调制器框图

Figure3-4 The modulation diagram of double-chaotic CSK

解调方式也是分为相干和非相干两种。图 3-5 为两个混沌发生器组成的 CSK 相干解调框图[43-44]。

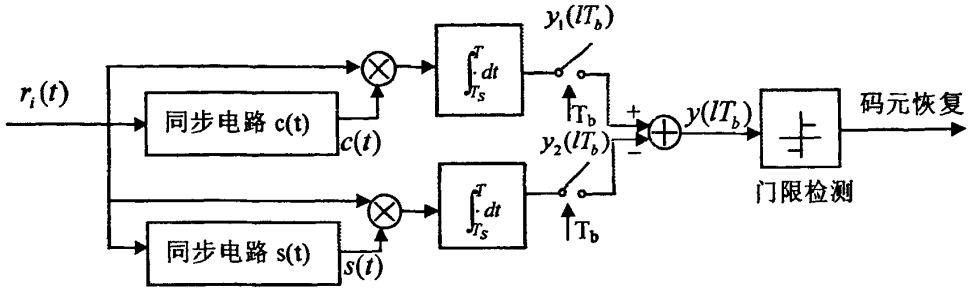


图 3-5 双混沌系统 CSK 相干解调电路

Figure3-4 The coherent demodulation circuit of double-chaotic CSK

在该解调方式中，采用了两个相关器对接收到的信号以及重构恢复的混沌信号进行相关检测。第 l 符号位的相关器输出值如式（3-7）所示：

$$y_1(lT_b)=\int_{(l-1)T_b+T_s}^{lT_b} r(t)c(t)dt$$
$$y_2(lT_b)=\int_{(l-1)T_b+T_s}^{lT_b} r(t)s(t)dt \qquad (3\text{-}7)$$

阈值检测器的输入值通过下式计算得到：

$$y(lT_b)=y_1(lT_b)-y_2(lT_b) \qquad (3\text{-}8)$$

如果第一个相关器的输出比第二个大，则判断为“+1”；否则即为“-1”。

由于相关检测需要一个同步的过程，在同步性较差的环境下，很难甚至不可能重构相同的混沌信号。因此非相干检测方法被认为是更加实用的方式。

最简单的非相干检测方法是基于 Kolumbán 等人于 1998 年提出的位能量检测机制[43]。该方法就是图 3-3 所示的 COOK 解调方式。假设对应于 $c(t)$和$s(t)$的平均位能量不同，分别以 E_{b1}和E_{b2}来表示。在无噪环境下，第 l 符号位的位能量检测器的输出为：

$$y(lT_b)=\begin{cases}\int_{(l-1)T_b}^{lT_b} r^2(t)dt, & \text{发送的符号位为}+1\\ \int_{(l-1)T_b}^{lT_b} s^2(t)dt, & \text{发送的符号位为}-1\end{cases} \tag{3-9}$$

与 COOK 的解调方式相似，检测器的阈值设为 E_{b1}和E_{b2}的均值，也就是 $(E_{b1}+E_{b2})/2$。同样当在噪声环境下，就需要提高阈值的设置。

3.2.2 差分混沌键控（DCSK）调制

CSK 调制技术是基于位能量检测来进行解调的，相关器的检测阈值应低于噪声值。否则就会出现大量误判，导致解调错误。为克服这个缺点，提出了差分混沌键控调制 （DCSK：differential chaos-shift-keying)方案[38]。图 3-6 和 3-7 就是 DCSK 调制与解调原理图。

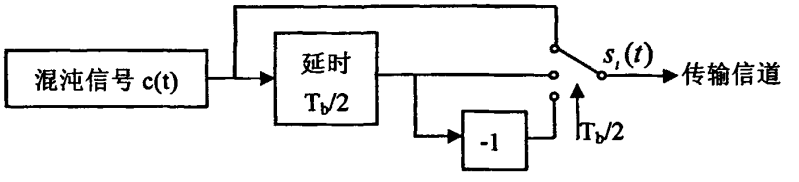


图 3-6 DCSK 系统调制电路框图

Figure3-6 The modulation diagram of DCSK

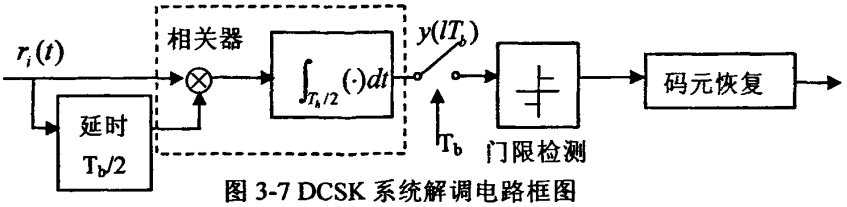


图 3-7 DCSK 系统解调电路框图

Figure3-7 The demodulation diagram of DCSK

DCSK 调制中，每个发送符号时间段都被划分为两个相同的时隙。第一个

时隙用来传输一个混沌参考信号，而第二个时隙用来发送一个承载码元信息的信号。在第二个时隙中，如果发送的符号为“+1”，则重复发送参考混沌信号；如果为“-1”，则发送一个相位相反的混沌信号。因此，发送的信号可表示为如下形式：如果第 l 符号位是“+1”，则

$$s(t)=\begin{cases}c(t) & (l-1)T_b \le t < (l-1/2)T_b \\ c(t-T_b/2) & (l-1/2)T_b \le t < lT_b\end{cases} \tag{3-10}$$

如果如果第 l 符号位是“-1”，则

$$s(t)=\begin{cases}c(t) & (l-1)T_b \le t < (l-1/2)T_b \\ -c(t-T_b/2) & (l-1/2)T_b \le t < lT_b\end{cases} \tag{3-11}$$

这里 $c(t)$ 代表混沌参考信号。

在接收端，在后半个符号时期内接收的信号将与前半个时期的信号进行相关运算。相关器的第 l 个符号周期的输出值可以表示为：

$$\begin{aligned}y(lT_b) &= \int_{(l-1/2)T_b}^{lT_b} r(t)r(t-T/2)dt \\ &= \int_{(l-1/2)T_b}^{lT_b} [s(t)+\xi'(t)][s(t-T_b/2)+\xi'(t-T_b/2)]dt \\ &= \int_{(l-1/2)T_b}^{lT_b} s(t)s(t-T_b/2)+\int_{(l-1/2)T_b}^{lT_b} s(t)\xi'(t-T_b/2)dt \\ &+ \int_{(l-1/2)T_b}^{lT_b} \xi'(t)s(t-T_b/2)+\int_{(l-1/2)T_b}^{lT_b} \xi'(t)\xi'(t-T_b/2)dt\end{aligned} \tag{3-12}$$

在无噪声干扰的环境下，$y(lT_b)$ 可以表示为

$$\begin{aligned}y(lT_b) &= \int_{(l-1/2)T_b}^{lT_b} s(t)s(t-T_b/2)dt \\ &= \begin{cases}\int_{(l-1)T_b}^{(l-1/2)T_b} c^2(t)dt & \text{发送的符号位为+1} \\ -\int_{(l-1)T_b}^{(l-1/2)T_b} c^2(t)dt & \text{发送的符号位为-1}\end{cases}\end{aligned} \tag{3-13}$$

因为 $c^2(t)>0$，当第 l 位符号为“+1”时，$y(lT_b)$ 是正值。当第 l 位符号为“-1”时，$y(lT_b)$ 是负值。当不考虑噪声影响时，检测阈值应设置为 0；如果加入了噪声，则对输出 $y(lT_b)$ 输出判决会有些影响，可能会导致其分布交叠处的信号

的误判。但从实验结果来看，DCSK 具有较好的抗噪声性能，能够对信道中的畸变具有一定的鲁棒性。

3.3 混沌同步保密通信仿真

3.3.1 混沌遮掩同步保密通信仿真

混沌遮掩通信方案主要应用了驱动－响应耦合混沌系统渐近同步的设计思想。以 Lorenz 系统为例，混沌遮掩保密通信系统结构如图 3-8 所示。

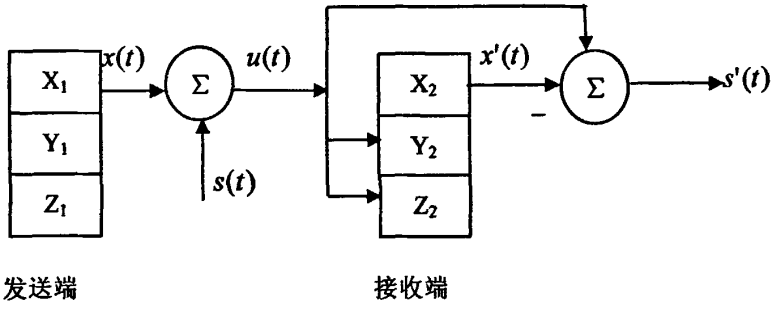


图 3-8 混沌遮掩保密通信系统结构图

Figure3-8 The structure diagram of chaos masking secure communication

该遮掩系统中应用的是相加的方式将有用信号与混沌信号进行混合[29]。发送方利用混沌 Lorenz 系统产生混沌信号 $x(t)$（我们选择的是输出端 X），与待传输的有用信息信号 $s(t)$ 相叠加，信道中传输的信号为 $u(t)=x(t)+s(t)$。接收方的混沌系统受响应 $u(t)$ 的驱动，收端混沌系统与发端混沌系统达到同步，重构出混沌信号 $x'(t)$，通过 $u(t)-x'(t)=s'(t)$ 得到有用信号。

仿真实例：

本设计采用的混沌遮掩方式是将有用信号与混沌载波信号相叠加，在叠加之前，又使用一个线性映射对有用信号进行变换，这样系统的安全性不仅依赖于混沌遮掩，还依赖于线性映射，即使攻击者截获了信道上的传送信号，也无法分析出有用信号，增加了信号被截获后的破译难度，这点对信息传输的安全性很有利。

设有用信号为 $s(t)=\sin(x)$，仿真方案在叠加 $s(t)$ 与混沌信号 $x(t)$ 之前将有用信号经过函数 $f(x)=x^3+1$ 变换，生成 $s_1(t)$ 再和混沌信号加性遮掩，产生信道信号 $u(t)$，经由信道传输。选择 Lorenz 系统作为驱动系统，如式（3-14）所示，Chen 系统为响应系统，如式（3-15）所示，同步以后的响应系统如（3-16）式所示。接收方的同步后的混沌信号为 $x'(t)$，$x'(t)$ 与 $u(t)$ 进行加性运算得到函数变换后的有用信号 $s_1'(t)$，再对 $f(x)$ 求反函数 $f^{-1}(x)=\sqrt[3]{f(x)-1}$，最终接收方得到的有用信号为 $s'(t)=\sqrt[3]{(u(t)-x'(t))-1}=\sqrt[3]{s_1'(t)-1}$，发送方有用信号与接收方的信号之间的误差为 $e(t)=s'(t)-s(t)$。Lorenz 方程和 Chen 方程分别作为驱动和响应系统，由混沌同步方法得到误差方程，式（3-15）中的 u1,u2,u3 就分别为调整误差的参数，经过计算得到的响应方程（3-16）。

$$\begin{cases}\dot{x}_1=10(y_1-x_1)\\ \dot{y}_1=28*x_1-y_1-x_1*z_1\\ \dot{z}_1=x_1*y_1-(8/3)*z_1\end{cases} \tag{3-14}$$

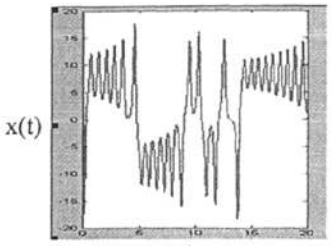
$$\begin{cases}\dot{x}_2=35(y_1-x_1)+u_1\\ \dot{y}_2=-7*x_2+28*y_2-x_2*z_2+u_2\\ \dot{z}_2=x_2*y_2-3*z_2+u_3\end{cases} \tag{3-15}$$

$$\begin{cases}\dot{x}_2'=10(y_2-x_2)\\ \dot{y}_2'=35*x_1-y_2-x_1*z_1-7x_2\\ \dot{z}_2'=x_1*y_1-(8/3)*z_2\end{cases} \tag{3-16}$$

系统的初始值：Lorenz 驱动系统的初始值为(-10,-20,15)，Chen 响应系统的初始值为(1,2,3)，仿真结果如图 3-9 所示。

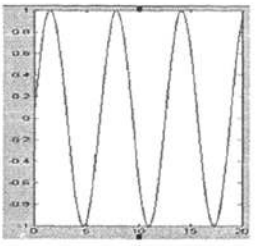
通过仿真结果，我们可以发现，利用这个系统进行的混沌遮掩，还原信号大概在 1s 后与输入的有用信号实现同步，有用的正弦信号经过线性变换后与混沌系统的 x 轴输出混沌信号进行的是加性运算，对有用信号进行遮掩，图 d 是信道中传输的信号，即变换后的有用信号和混沌信号叠加的和信号，

从时域图上可以看出该正弦有用信号已经被混沌信号成功掩盖，无法辨识。



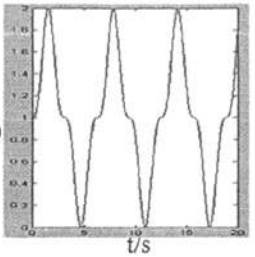
a) Lorenz 驱动信号

a) Lorenz driver signal



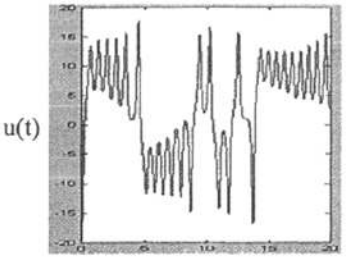
b) 有用信号

b) useful signal



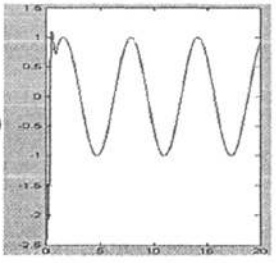
c) 经变换后的有用信号

c) transformed useful signal



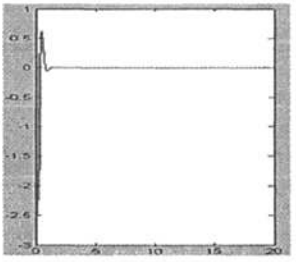
d) 信道中传输的信号

d) transmissive signal in channel



e) 还原后的有用信号

e) restored useful signal



f) b 与 e 图的误差曲线 e(t)

f) error curve e(t) between b and e

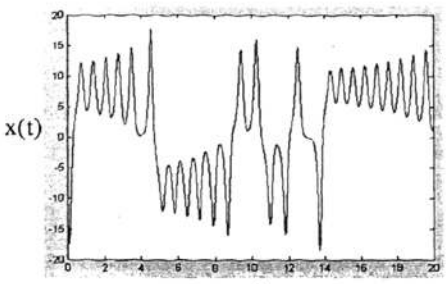
图 3-9 传输有用信号为 sin(t)时的仿真图

Figure3-9 The simulation figure when transitting useful signal sin(t)

从图中我们可以看出，待传输的有用信息的能量要远远小于 Lorenz 载波信号的，那么如果传输的有用信号能量比较大的时候，结果如何呢？

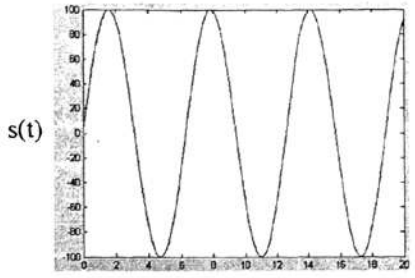
仍然利用上述模型，当混沌载波遮掩大信号（被遮掩信号为混沌载波信号约 5 倍时），传输有用信号能量较大时的仿真结果如图 3-10 所示。

仿真说明当传输的有用信号能量较大时，载波混沌信号的波形上就将泄漏出原始信息的特征，因此也就失去了保密功能。因此该混沌遮掩保密通信实现了首发方的同步，但其保密性却受到待传输有用信号的能量限制，能量较大时丧失了保密性。



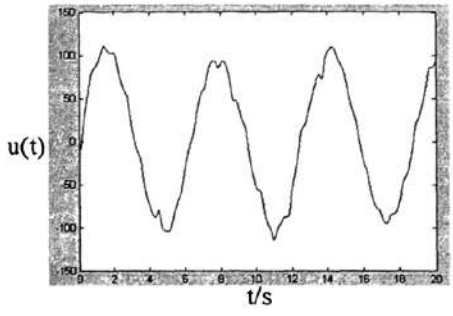
a) 混沌载波信号

a) chaotic carrier signal



b) 被遮掩的大信号

b) masked large signal



c)信道中传输信号

c) transmissive signal in channel

图 3-10 传输信号能量较大时的仿真图形

Figure3-10 The simulation figure when transitting large signal

3.3.2 混沌键控同步保密通信仿真

1、CSK 相干解调方式仿真

如图 3-11 所示，在这里我们仿真的是采用两个混沌发生器的 CSK 调制解调系统。

发送端所采用的混沌方程为蔡氏混沌电路，也称 Chua 电路，它是蔡绍堂教授于 1983 年提出的第一个混沌模拟电路。经过数学分析后，由归一化得到其数学方程为：

$$\begin{cases} \dfrac{dx}{d\tau}=\alpha[y-x-f(x)] \\ \dfrac{dy}{d\tau}=x-y+z \\ \dfrac{dz}{d\tau}=-\beta y \end{cases}, \text{其中 } f(x)=\begin{cases} bx+a-b, & x\geq 1 \\ ax, & |x|\leq 1 \\ bx-a+b, & x\leq -1 \end{cases} \qquad (3\text{-}17)$$

已知取 $\alpha=10$, $\beta=15.68$, a = -1.2768, b= -0 .6888 这组参数时，Chua 混沌电路能够形成双螺旋吸引子。

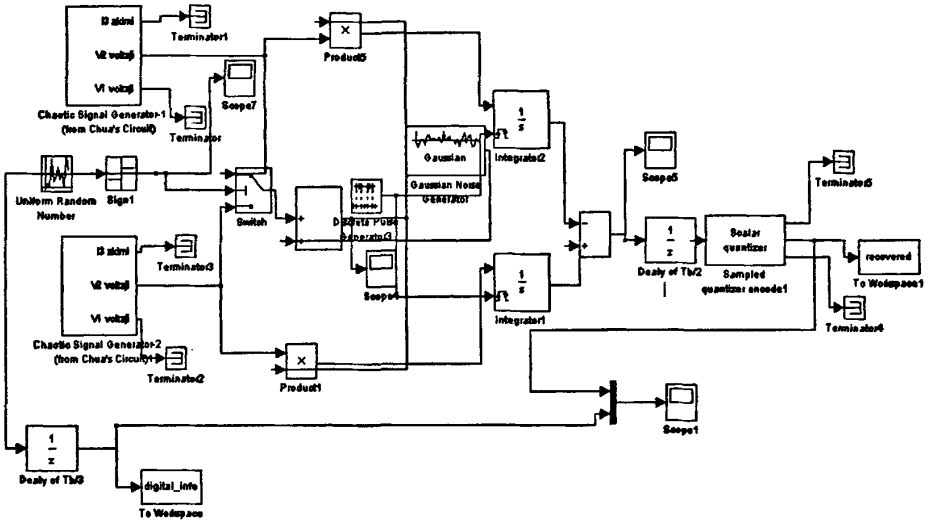


图 3-11 相干 CSK 调制解调系统框图

Figure3-11 The block diagram of coherent CSK modulation and demodulation system

图 3-11 利用 simulink 搭建了相干 CSK 调制解调系统。发送的信息信号由随机数产生以及过零比较模块产生“+1”和“-1”信号；发送端根据传输符号“+1”和“-1”来通过选择器模块相应的传送混沌信号 $c(t)$和$s(t)$，$c(t)$和$s(t)$采用的是 Chua 混沌电路；信道采用的是高斯白噪声信道；接收端通过乘法器和积分器构成相关器实现相关运算，对接收到的信号以及重构恢复的混沌信号进行相关检测；再通过阈值检测器进行差值比较，如果第一个相关器的输出比第二个大，则判断为“+1”；否则即为“-1”。

采用的是前面所讲的相干解调方式，经过比较电路的输出信号经过采样量化输出，调制信号与解调后的信号仿真结果如图 3-12 所示，从可以看出，经过相干解调正确的恢复了发送信号。

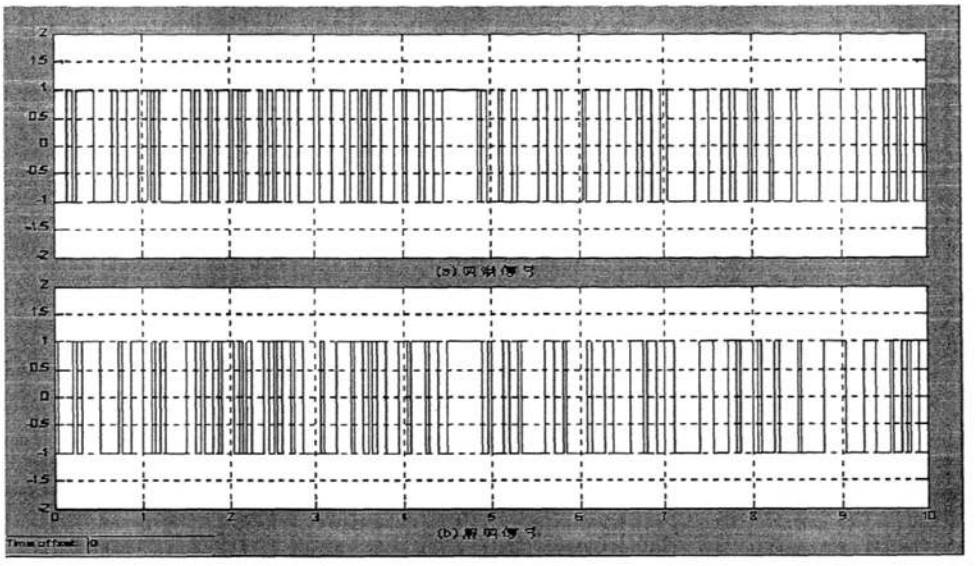


图 3-12 相干 CSK 仿真波形图

Figure3-12 The simulation waveform figure of coherent CSK

2. CSK 非相干解调方式仿真

CSK 非相干解调系统框图如图 3-13 所示。

从设计的框图中可以看到，非相干 CSK 的发送端信息信号同样由随机数产生以及过零比较模块产生“+1”和“-1”信号；发送端根据传输符号“+1”和“-1”来通过选择器模块相应的传送混沌信号 $c(t)$和$s(t)$，$c(t)$和$s(t)$采用的是 Chua 混沌电路；信道采用的是高斯白噪声信道；二者的区别在于接收端利用乘法器和积分器组成一个位能量估值器，并通过一个阈值检测器进行比较，如果估算位的能量大于这个阈值，则解码为“+1”；否则为“-1”。

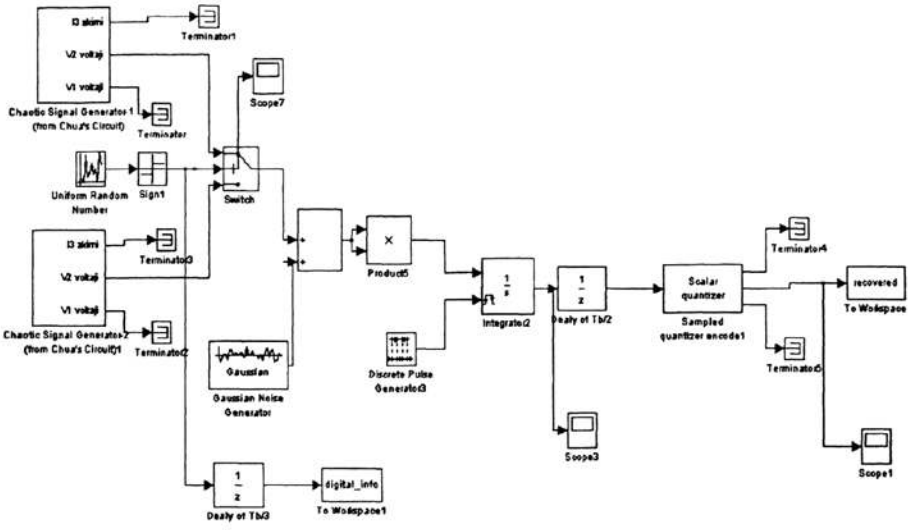


图 3-13 非相干 CSK 调制解调系统框图

Figure3-13 The block diagram of noncoherent CSK modulation and demodulation system

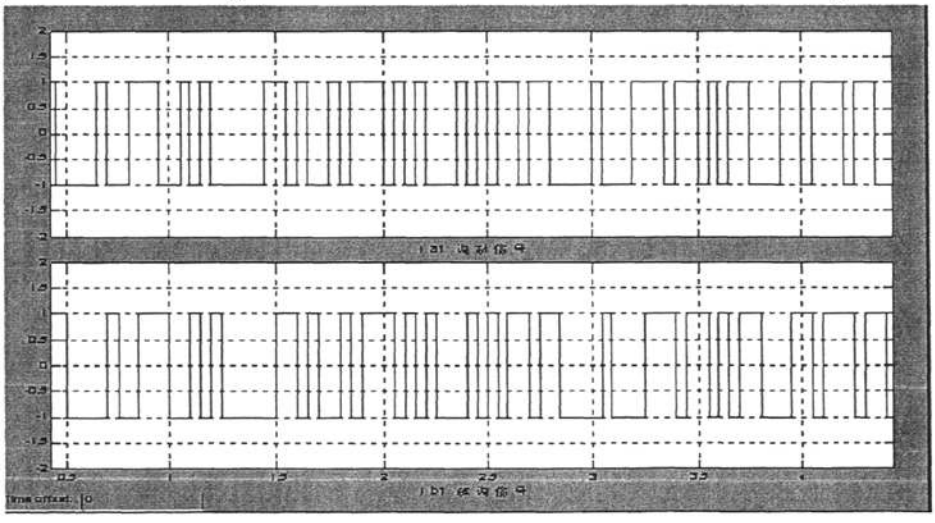


图 3-14 非相干 CSK 仿真波形图

Figure3-14 The simulation waveform figure of noncoherent CSK

这里采用的是非相干解调方法。从两种解调系统的对比中即可发现，非相干解调方法不需要在解调端重构两个混沌系统，因此避免了混沌系统精确同步难于实现的弊端。调制信号与解调后的信号波形图如图 3-14 所示，从图中可以看出经过非相干解调也正确的恢复了发送信号。

3.DCSK 仿真

图 3-15 为设计的 DCSK 调制与解调仿真框图。

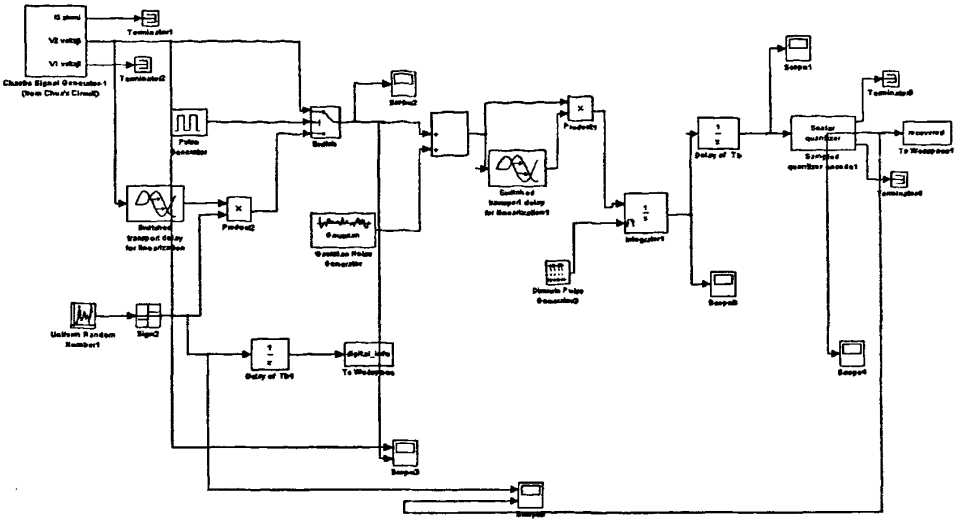


图 3-15 DCSK 调制与解调系统框图

Figure3-15 The block diagram of DCSK modulation and demodulation system

从图中可以看出，发送端载波信号为 Chua 混沌信号，仿真系统时间为 10s，DCSK 系统用两种混沌信号来传输信号。本仿真的信息信号的符号周期为 $T_b=0.05$，仿真中采用的群延迟模块起到延迟 $T_b/2$ 的作用。在符号周期的前半周期内，产生一组混沌信号作为参考源，在后半周期再重复产生一段混沌序列，以此来调制信息信号。如果经开关输入的为“+1”，则后半周期传送与参考信号同相的混沌信号；若为“-1”，则传送与参考信号反相的混沌信号。同样采用的是高斯白噪声信道进行传输，接收端将接收到的混沌叠加信号与延迟混沌信号进行非相干解调（与 CSK 非相干方式类似），最终得到解调输出结果。

DCSK 系统发送端调制信号与接收端解调信号的波形图如图 3-16，从图中可以看到 DCSK 正确完成了信号的调制与解调。在改变噪声的干扰值时我们发现噪声能量很大时会带来接收端的误判，误码率将提高，通过提高信息

信号的能量可以降低误码率；如果提高延时时间T_b，则可以延长相关器积分时间，这样也可以使接收端采样判决的准确性提高，从而降低误码率。但是积分时间变长，会直接导致信息信号传输时间延长，影响整个系统的传输速率。

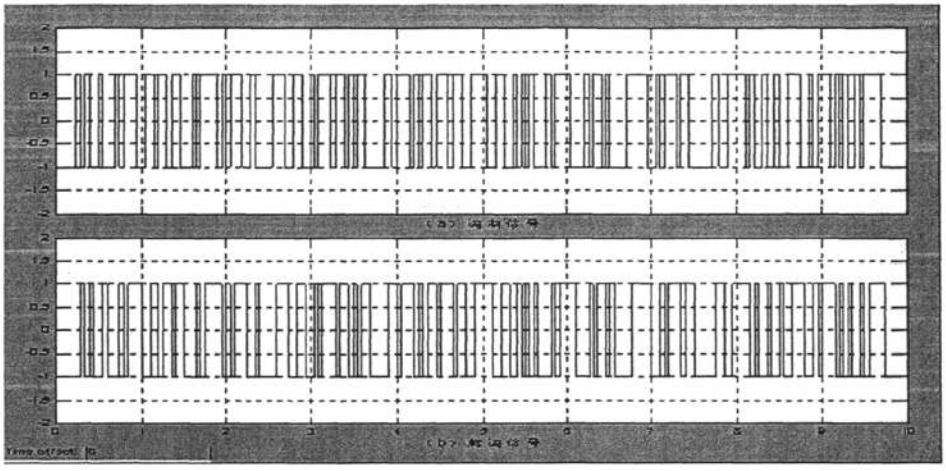


图 3-16 DCSK 仿真波形图

Figure3-16 The simulation waveform figure of DCSK

3.4 本章小结

本章主要研究了与实现了混沌同步保密通信理论。从理论与实验角度分别分析了混沌遮掩与混沌键控保密通信技术。

首先介绍了混沌遮掩保密通信的方法与原理框图，分析了加性，乘性以及加乘结合的几种遮掩方法。然后分别分析了 CSK 以及 DCSK 调制与解调方法，给出其原理框图，并从理论上分析了相干以及非相干两种解调方式。最后通过 Matlab 编程以及 Simulink 建立模型，通过仿真实验分别验证了 CSK 以及 DCSK 系统的有效性。通过理论研究与仿真分析，我们可发现：

混沌遮掩保密通信要求有用信号信息的能量必须要小于混沌载波能量，一旦大于它，遮掩将失去保密性，从而造成信息泄露。

相较于混沌遮掩系统，混沌键控技术的抗噪声以及参数失配的能力较强，

可应用于数字通信，而且混沌键控通信系统克服了混沌遮掩通信方式中要求混沌载波功率大于有用信号的缺点。

CSK 混沌移位键控系统有相干和非相干两种解调方式。相干解调需要在接收端重构发送端的混沌信号源，因此对混沌系统的同步性能要求很高，而且要依赖于混沌信号的能量，同步很难实现而且信号能量一旦变化都很容易引发误判，使系统传输误码率提高。

DCSK 的优点是解调不需要混沌信号的再生，直接通过对前半信号周期混沌参考信号的延迟与后半信号周期载波调制信号的相关运算即可实现码元判决，结构比较简单且容易实现。

DCSK 的缺点在于它只能应用于低码率的混沌数字保密通信中， 因为从设计原理上我们可以看到 DCSK 在传送每个信息信号的同时需要先发送混沌参考信号，因此系统传输带宽以及速率上欠佳。

无论是 CSK 或是 DCSK 通信系统，其保密性及鲁棒性不甚理想，这两种混沌系统都比较容易被破译。它们共同存在的问题是:为使系统正常工作，接收的二进制数字信号的误码率较低，码元宽度必须限制在毫秒级内。因此，它们也只能用于较低速码率的混沌数字保密通信中。为解决这个问题，目前对混沌键控保密通信的研究有了更加新颖与广泛的技术。比如调频差分混沌移频键控(FM-DCSK)以及四相正交混沌移频键控(QCSK)等等。

第4章 混沌序列密码算法及自同步技术的研究实现

由于混沌具有的遍历性、混淆性、精确性以及对初始条件敏感的很多基本特性，混沌理论已经被很多学者将其与具有扰乱和扩散特性的传统密码学结合起来进行研究，利用混沌理论来丰富密码学理论。从1989年开始，人们已经先后设计和分析了多种混沌密码算法。使用混沌加密的想法可以追溯到香农出版在1949年的经典论文"保密系统的通信理论"[12]，Ljupĉo Kocarev[45-46]等人已经证明了如何利用混沌构造一种类似DES的分组密码。近年来，已有Ljupĉo Kocarev[47]，Jesús Urías[48]以及李树钧[49]等人研究利用混沌来设计S盒以及一些新型密码算法，以上这些文献无不表明混沌理论可以采用类似于密码设计者设计传统常用密码算法的方法来设计混沌密码算法。这些研究显示混沌可以象传统密码学中一些常用基本算法一样很自然地在传统密码学的框架下得到运用。

另外，由于通过确定性的密码系统经特定变换产生伪随机输出序列具有良好的密码特性，例如分组密码中经常用到的取模操作和非线性S盒使有意义的明文变成"乱码"似的密文[50]。因此从算法设计来看，任何好的密码系统也可以当做是一个混沌或者拟混沌系统[51]。传统的密码系统将明文加密为混乱的密文，与复杂动力系统产生的混沌现象非常类似。在文献[52]中，Marco Götz等人的研究表明在某些传统流密码中存在着（伪）混沌行为。实际应用中，当混沌系统在计算机有限精度下实现时，传统密码学中常用的模（mod）运算也经常被采用来对混沌方程的离散形式进行描述[53-54-55]。因此我们有理由相信混沌密码学的研究有助于丰富传统密码学的内容，为好的密码系统的设计提供更多的设计思路和手段。

混沌密码学的研究主要有两个基本模式：第一个模式是模拟混沌保密系统，它们一般是在模拟电路系统中基于混沌同步技术实现的[30]；第二个模式

则对应在计算机（或者数字电路）上实现的数字化混沌密码设计，利用混沌系统构造新的流密码和分组密码，主要基于计算机有限精度下实现的数字化混沌系统，它们和混沌同步技术无关。一般而言，基于同步的混沌保密系统是设计用来实现有扰信道上的保密通信，不能直接推广到传统密码学中去。更为重要的一点是，大量的密码分析工作已经表明大部分基于混沌同步的保密通信系统都不够安全，攻击者可能从截获的信号中恢复和秘密参数相关的部分有用信息（甚至可能近似确定这些参数）[56-60]。因此，尽管混沌同步技术目前仍然在保密通信领域得到广泛的研究，但是相关的成果对于传统密码学而言价值比较小。本章将主要研究离散混沌系统—即数字化混沌流密码的设计与应用。

4.1 数字化混沌密码理论

Robert A.J.Matthews 于 1989 年发表的文章中第一次明确提到“混沌密码”并得到广泛关注[61]，该文提出了一种基于变形 Logistic 映射的混沌流密码方案。自从 Matthews 的流密码提出以来，数字混沌密码受到了来自不同领域研究者越来越多的关注。在其之后的几年间不断有多篇关于数字化混沌密码的设计与分析的论文发表。1997 年以后，一些新的数字化混沌密码的提出开启了新一轮的研究热潮。例如周红、凌燮亭等人提出的“离散均匀输入”+“逐段线性混沌映射”混沌流密码方案[60]；Yen J.C.与 Guo J.I.提出的三类混沌图象加密方案[61]；易开祥、孙鑫、石教英等人提出的基于混沌序列的图象加密算法[62]以及 E.Alvarez 等人提出的混沌加密新方法[63]。通常来讲，数字化混沌密码的设计有两种方案：第一是利用混沌系统生成直接用于掩盖明文的伪随机密钥序列；第二是使用明文和密钥作为初始条件以及控制参数，通过反复多次迭代的方法得到密文。第一种方案对应传统流密码，第二种则相应为分组密码。除了以上两种之外，最近几年又出现一些新的设计思路，比如在分组密码中使用混沌生成 s 盒[45-46-48]以及采用搜索机制的数字化混沌密码设计[66-69]。从以

上的研究成果中看，目前大多数数字混沌密码的研究都是基于理论上的研究，采用软件模拟仿真而实现的。我们则依托国家自然科学基金项目“基于 FPGA 技术的混沌序列密码加密芯片的研究（60672011）”开展了数字化混沌密码研究的应用领域的设计和探讨。以下将介绍我们研究的数字化混沌加密算法的实现。

4.2 数字化混沌流密码算法的研究与实现

4.2.1 A5/1 流密码算法原理

A5/1 算法框图如图 4-1 所示。

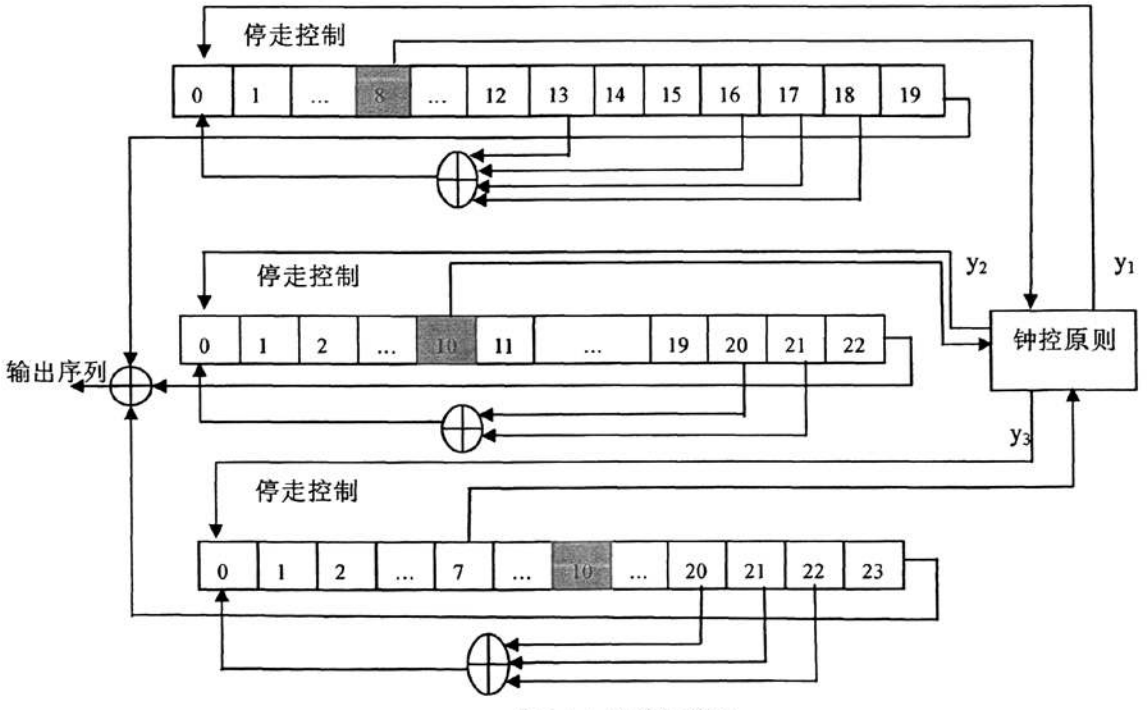


图4-1 A5/1算法框图

Figure4-1 A5/1 arithmetic block diagram

A5 流密码算法是一个用于数字蜂窝移动电话 GSM 加密的序列密码标准。它的加密效率非常高，能通过所有已知的密码学统计测试，并且具有适合硬

件实现、加密延迟时间短等优点。第一个“原始”的 A5 算法被重命名为 A5/1。A5 算法还包括 A5/0—意味着没有加密；A5/2—较弱的空中加密算法；A5/3—已经利用到无线通讯领域中。大多数 A5/x 算法从时间复杂度上来讲都要弱于 A5/1[70]。

A5/1 是一种典型的基于 LFSR 的流密码算法，它是集 LFSR 与钟控于一体的算法。算法中含有 3 个线性反馈移位寄存器 LFSR，详细说明如表 4-1：

表 4-1 A5/1 的 LFSR 的选择

Table4-1 The LFSR choice of A5/1

LFSR	存储单元	本原多项式	时钟控制位	抽头位
1	19	$x^{18}+x^{17}+x^{16}+x^{13}+1$	8	13,16,17,18
2	22	$x^{21}+x^{20}+1$	10	20,21
3	23	$x^{22}+x^{21}+x^{20}+x^{7}+1$	10	7,20,21,22

A5/1 算法的 3 个 LFSR 采用大数原则来控制停走模式进行钟控，钟控位置均在 LFSR 的中间抽头上，依次为第 8 抽头、第 10 抽头和第 10 抽头。表 3-2 所示为钟控函数的规则：

表 4-2 钟控规则

Table4-2 The clock control rules

f(a(t+11),b(t+12),c(t+13)) = (y1,y2,y3)	a(t+11) b(t+12) c(t+13)
(1,1,1)	0 0 0 1 1 1
(1,1,0)	0 0 1 1 1 0
(0,1,1)	0 1 1 1 0 0
(1,0,1)	1 0 1 0 1 0

a、b、c 为三个 LFSR 抽头的状态值，钟控函数表达式为 clk=ab+ac+bc，当 a、b、c 与 clk 的值相同时，驱动 LFSR 输出序列，否则停止，同一时刻至少有两个 LFSR 工作。三个 LFSR 的序列输出再进行异或得到最终输出序列值。

从目前密码分析文献中已经获知，针对 A5/1 的密文攻击方法已经被大量

的证实。但是 A5/1 算法以其加密效率高、易于硬件实现的优势仍在广泛的使用。我们也将利用它的优势对其进行改造设计出保密性更强的基于 A5/1 与 logistic 算法的数字混沌密码算法。

4.2.2 离散 logistic 混沌加密算法设计

Logistic 混沌映射的迭代方程为

$$x_{n+1}=\mu x_n(1-x_n) \quad \mu\in(0,4]\,;x_n\in(0,1) \qquad (4\text{-}1)$$

通过设置μ参数和初始值 $x(0)$ 使之产生混沌信号。在上一章通过对其特性的分析我们选取了 $\mu=4$ 作为其参数，此时可确定 logistic 映射处于混沌区间内。

因为要在硬件电路中实现混沌加密算法，因此我们需要将原连续的混沌映射输出进行离散二值化。为此我们采用 matlab 中 dspbuilder 模型转化成 vhdl 语言的设计方法设计了离散化方案[71]，成功搭建 logistic 数字化加密模型，并将其转化为 vhdl 语言，下载到芯片中进行加密测试，缩短了其硬件开发周期。

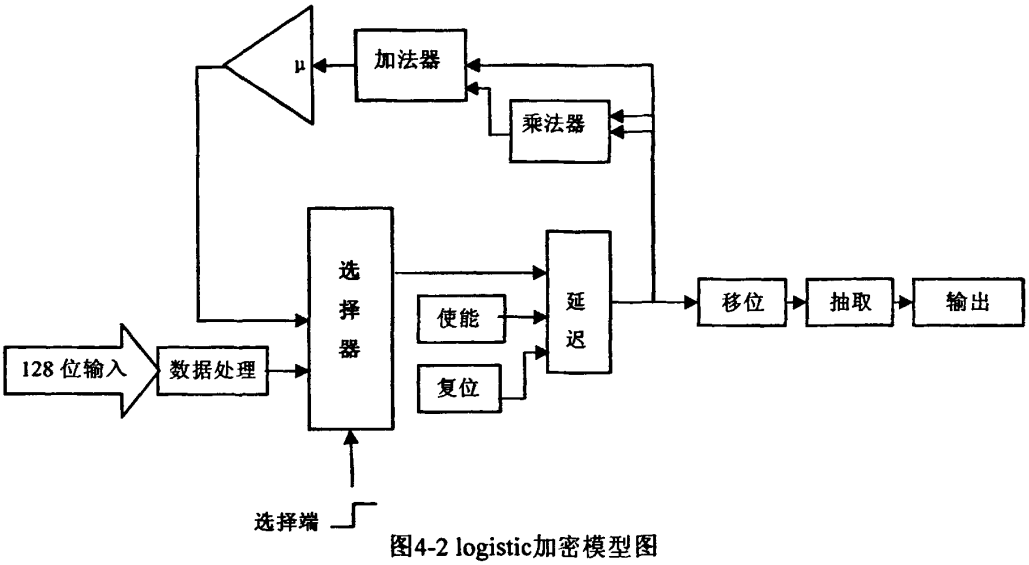


图4-2 logistic加密模型图

Figure4-2 The encryption model diagram of logistic map

一、logistic 加密模型图的设计

考虑到高保密性的要求，为使接口满足能够进行 128 位二进制的输入，并在此基础上保证输入要求，我们在 dspbuilder 下建立以下模型，输出为满足一定随机性要求的 0-1 序列。模块原理图如图 4-2。

原理图包括以下几部分：

（一）模型接口部分设计

根据设计需要，输入满足初始值在 0-1 范围之间随机给定的 128 位二进制输入。

1、128 位输入设计问题

由于 MATLAB 所支持最大输入位宽为 51 位，所以，要求 128 位输入，必须进行一系列的合并与运算。步骤如下：

(1)我们将 128 位输入，每十六位分成一组，一共分为八组，分别为 Input1-Input8。

(2)分别将 Input2 与 Input3，Input4 与 Input5，Input6 与 Input7，进行与操作，至此输入变成五组。

(3)再通过 Bitwise Logical Bus Operator 模块将其与 Input8 相互进行与操作，变成两组 16 位输入，通过 Bus Concatenation 模块使其合并成 32 位。

(4)最后将 Input1 与这个 32 位再经过 Bus Concatenation 合并，最终使其变成 48 输入，满足 MATLAB 可接受的位宽要求。

2、初始值满足 0-1 范围

由于 logistic 方程，初始值 x 的取值范围为 0-1 之间，于是通过接口输入的二进制值必须转化为处于 0-1 范围内的值。由于合并后的 48 位二进制输入范围为 $0\sim2^{48}$，所以，为了保证选择器初始输入值的范围在 0-1 之间，我们的方法是使用 Barrel Shiffer 模块进行二进制移位，向左移位 48 位即可。至此输入值满足 128 位接口以及 0-1 范围内的要求。

3、输入值选择

选取 n-to-1 Multiplexer 模块，在选择控制端为 1 时输入初始值，在控制端为 0 时，输入循环计算值。

（二）Logistic 方程循环计算部分

根据方程 $x_{n+1}=\mu x_n(1-x_n)$，将其化简为 $x_{n+1}=\mu x_n-\mu x_n^2$，于是使用 product，Parallel Adder Subtractor, 和 Gain 模块。当 μ=4 时，该映射是满射，产生的混沌序列在区间(0,1)上具有遍历性，于是连接图中 μ 值取 4。

（三）数据与位处理部分

1、重要模块 Altbus

该模块的作用主要有两个，将二进制转换为十进制，与控制位宽。设置的位宽数对产生的 0-1 序列周期有影响，位数越大，周期越大。

2、移位抽取

通过 Barrel Shiffer 将输出数值移位 9 位，即相当于将数值扩大 2^9 倍，再利用 ExtracBit 抽取小数点前最后一位的数值，使其为奇数值时输出 1，偶数值时输出 0，以此产生 0-1 随机序列。

二、Matlab/dspbuilder 设计流程

图 4-3 是利用 Dspbuilder 进行模型—硬件语言设计的流程框图[28]。

Dspbuilder 是一个系统级（或算法级）设计工具，它架构在多个软件工具之上，并把系统级（算法仿真建模）和 RTL 级（硬件实现）两个设计领域的设计工具连接起来，并且最大程度的发挥了两种工具的优势[56]。Dspbuilder 依赖于 Mathworks 公司的数学分析工具 Matlab/Simulink，以 simulink 的 Blockset 出现。通过该工具既可以在 simulink 中进行图形化设计和仿真，同时又通过 signalCompiler 模块把 Matlab/simulink 的设计文件（mdl）转成相应的硬件描述语言 VHDL 设计文件（vhd），以及用于控制综合和编译的 tcl 脚本。而对后者的处理可以由 FPGA/CPLD 开发工具 QuartusII 来完成。

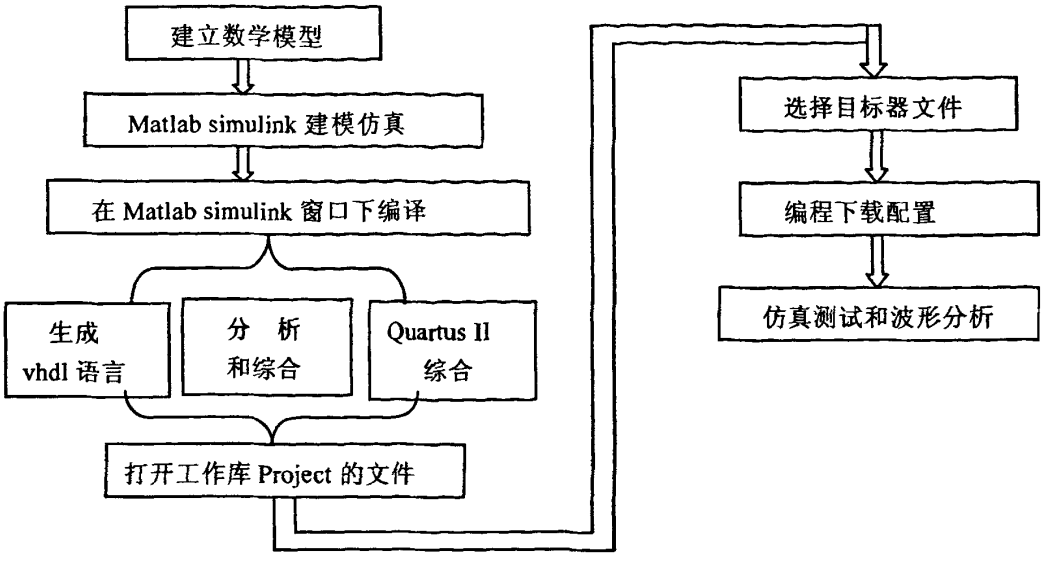


图 4-3 用 DSP Builder 工具开发设计步骤

Figure4-3 The develop and design steps of Dspbuilder

我们利用该方法得到了logistic加密模型的硬件设计电路，通过对各个输入引脚的初始化，我们在Quartus II下仿真得到了以下0-1二值序列仿真输出波形图4-4。

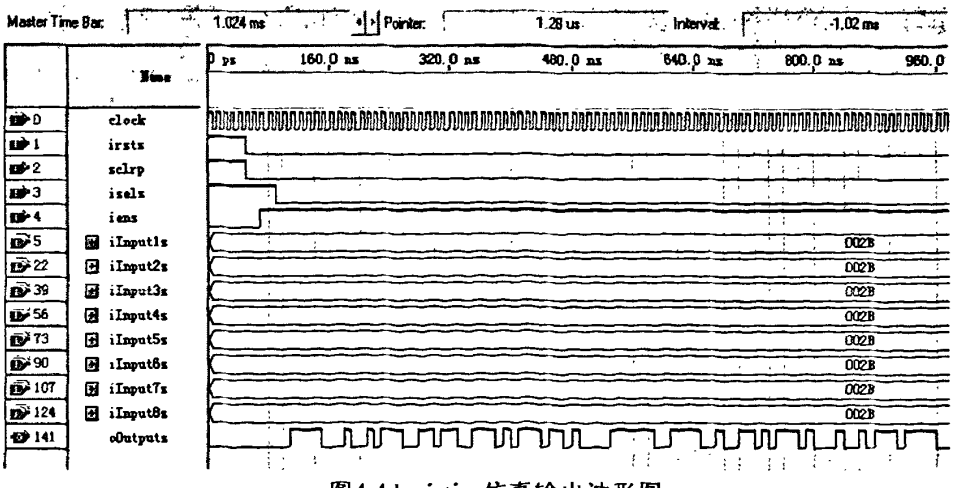


图4-4 logistics仿真输出波形图

Figure4-4 The simulation output waveform of logistic map

从图中看出经过我们的离散化设计，得到了 logistic 的 0-1 密钥序列输出，以下将会对输出的 0-1 序列进行统计学分析，包括平衡性以及游程特性，以此

来检验得到的输出序列是否具备较好的密码特性。

4.2.3 离散 logistic 混沌序列特性分析

在波形仿真成功后，我们将其下载到FPGA至ACEX1K-EP1K30TC144-3芯片中，使用安捷伦1693A逻辑分析仪观测系统硬件实现波形，以对输出的二值序列进行特性分析[28]。通过逻辑分析仪得到的输出序列的波形，可以用CSA格式存储并在Matlab中运用csvread命令将逻辑分析仪所得的结果调用并进行序列特性分析和测试。Matlab调用数据的命令格式为csvread('data.xls')。

1、平衡性检验

Logistic模型类似于一个随机系统，它的输出序列满足均匀分布的特性。因此，验证数字化模型的输出序列的统计特性，即序列的0与1数目是否趋于平衡。

不平衡度的计算公式为：
$$\frac{(Q_1-Q_0)}{N}\% \qquad (4\text{-}2)$$

其中Q_1为待测序列中数字1的个数，Q_0为待测序列中数字0的个数，N为待测序列总的数字个数。取7组不同序列长度离散混沌序列数据进行检验，经过统计测试得出输出序列中的0/1分布趋于均衡，不平衡度的计算值均在$1/10^6$以内。

2、游程特性的分析

理论上,在一个长度序列中取值相同的那些相继的元素合称为一个游程，该测试的目的是为了判断序列中不同长度的游程是否具有随机序列的游程特性。结果表明实际与理论的游程性质相接近，即序列中长度为1的游程数目约占总游程数的1/2，长度为2的游程数目约占总游程数的$1/2^2$，……，长度为k的游程数目约占总游程数的$1/2^k$，依次递减。而且任意长度0的游程个数和1的近似相等。经过对我们得到的输出序列的测试，其游程分布均满足$1/2^k$的理论结果。

以上特性分析可验证输出二值化序列的统计特性接近随机序列分布。接下来将利用logistic混沌方程对图像进行加密测试，并将演示加密对初值敏感性的体现。

4.3 混沌算法加密测试

加密测试结果如图 4-5 所示。

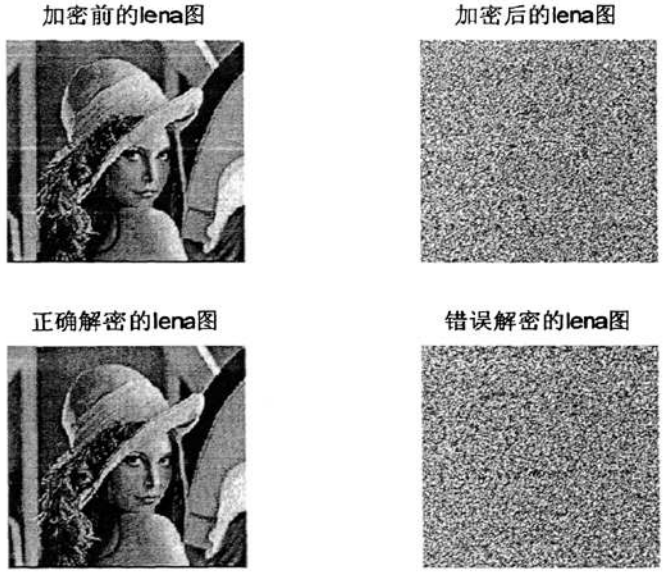


图 4-5 混沌算法加密测试

Figure4-5 chaotic arithmetic encryption test

所采用的加密图像为256* 256像素的‘lena图’，任选初值为 $x_0 = 0.123456$，将logistic混沌输出序列与‘lena’图按位进行异或运算可以得到加密后的图像；不改变 x_0 的初始值，进行反变换则可正确解密；但当我们稍微改变 x_0 的输入值，令 $x_0 = 0.1234567$ 后则无法正确解密。可以看到，微小的初值改变对解密产生了巨大的影响。

但在计算机仿真中，由于有限精度效应会带来离散混沌序列的混沌特性的退化，以下将就logistic离散混沌序列在有限精度下的周期特性进行分析。

4.4 有限精度下 logistic 混沌序列周期特性

在经典混沌理论中，所有的动力学系统都是定义在连续域上的，而它们的动力学特性往往只在具有正的Lebesgue测度的连续相空间中才有意义[70]。实际应用中在二进制数字化空间中的离散化混沌系统（即在2^{-L}—离散化空间中的混沌系统，这里L是计算机实现时的有限精度）被认为是数字化混沌系统（digital chaotic system)。当混沌系统在计算机上实现时，由于有限精度和有限状态问题，数字化混沌系统与理想的连续混沌系统在动力学特性上存在相当大的差异(特性退化)。典型的问题包括：出现短周期、轨道分布退化和产生相关性[72]。这种退化对数字化混沌密码的安全有不可忽视的影响。在混沌理论界有关数字化混沌的退化现象被广泛地报道和研究[73-76]。假设混沌在2^{-L}离散化的数字化空间中实现，现在我们来分析下数字化混沌带来的几种影响因素。

首先，数字化混沌在每次的迭代中不可避免地要引入量化误差，这种量化误差会使得拟混沌轨道以一种非常复杂而不可预测的方式与连续系统下的实际混沌轨道发生偏离[77]。同时又由于混沌系统对初始条件极其敏感，从而使得有限精度下的拟混沌轨道在经过有限的几次迭代之后就会变得与真实轨道完全不同。

还有一个因素是不可避免的周期性[78]。因为数字化混沌迭代受有限精度的限制，其离散空间范围是2^L个元素，因此每条拟混沌轨道最后都会变成周期性的，也就是说，最终总要进入一个循环而该循环的周期必然不大于2^L。

图4-6是一个典型的拟混沌轨道的示意图。一般来说，每条拟混沌轨道包括两个部分：$x_0, x_1 \ldots, x_{l-1}$和$x_l, x_{l+1} \ldots, x_{l+n}$，分别被称为暂态过程（transient branch）和周期（cycle）。对应地，l和$n+1$分别被称为暂态长度（transient length）和循环周期（cycle period），$l+n$称为拟混沌轨道长度（orbit length）。

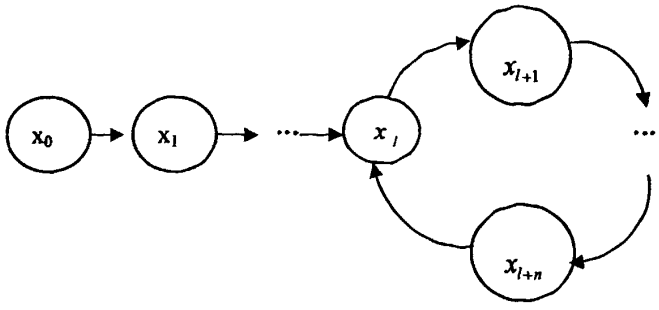


图4-6 拟混沌轨道的示意图

Figure4-6 The schematic diagram of pseudo chaotic orbit

在VC下利用编程模拟计算机有限精度效应来对logistic迭代序列进行了周期测试，循环周期查找的C语言计算过程如下：

（1）设置迭代方程：$x_{n+1}=4x_n(1-x_n)$；

（2）根据计算精度，定义初始值x_0，$if\,(r=1), x_0=0.1$；$if(r=2)$ $x_0=0.11$ .....

（3）在进入循环周期之前会出现一个暂态过程，因此设置一个较大的暂态迭代值，在迭代计算中始终保持精度为r。

（4）通过控制搜索步进值step，找到迭代序列的循环周期。

表4-3和图4-7为分别在十进制精度为1-15（即小数点后保留位数）下，比较了logistic迭代序列的周期与迭代初始值之间的关系。表中的r代表迭代精度，lastbit表示对初始值的微弱扰动，例如lastbit=1代表相应的初始值末尾数为1，lastbit=2代表相应的初始值末尾数为2，依次类推。在选定$\mu=4$的情况下经过对logistic方程$x_{n+1}=\mu x_n(1-x_n)$进行了15种精度控制，1~9种不同初值微扰，几十组的初值变化的统计平均，最终得到了受精度效应影响，logistic序列周期估计表4-3，并将结果绘制如图4-7。

表4-3 logistic序列周期估计表

Table4-3 The period estimated table of logistic sequence

r	lastbit=1	lastbit=2	lastbit=3	lastbit=4	lastbit=5	lastbit=6	lastbit=7	lastbit=8	lastbit=9
1	4	4	4	4	4	4	4	4	4
2	9	9	9	9	9	9	9	9	9
3	10	10	10	10	10	10	10	10	10
4	21	21	21	21	21	21	21	21	21
5	267	267	267	267	267	267	267	267	267
6	395	395	395	395	395	395	395	395	395
7	1955	1955	1955	1955	1955	1955	1955	1955	1955
8	659	3373	4225	2826	5136	2282	4382	157	741
9	18472	18472	18472	18472	18472	18472	18472	18472	18472
10	41056	28920	11165	75515	85746	37386	31308	12998	16691
11	196978	196978	196978	196978	196978	196978	196978	196978	196978
12	170628	116163	726836	365166	575459	40324	189734	252826	58137
13	671175	671175	671175	671175	671175	671175	671175	671175	671175
14	1279503	1279503	1279503	1279503	1279503	1279503	1279503	1279503	1279503
15	2222300	2222300	2222300	2222300	2222300	2222300	2222300	2222300	2222300

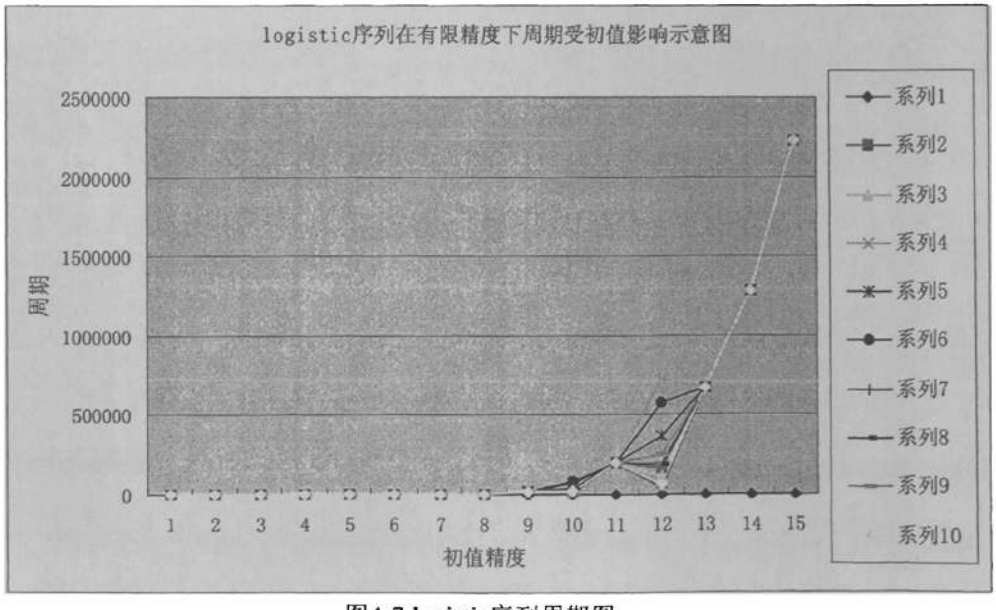


图4-7 logistic序列周期图

Figure4-7 The periodogram of logistic sequence

从图中我们可以看到，由于受到有限精度的限制，logistic序列值在不同精度下均有循环周期出现，但整体上看，随着精度的提高，周期也在增大；并且可以从图表中明显看出受精度效应影响，logistic混沌映射对初始条件的敏感性已经大大的退化。因此在选择logistic初始值的时候，应当提高其有效精度，使其达到最大，这样才能保证迭代序列的大周期。在我们设计的logistic加密方案中我们采用的初值精度为128bit的二值输入，经过处理后实际输入为48bit，因此其密钥空间范围可达2^{48}，从密钥空间角度来分析，具备良好的保密特性的，并且已经验证其具备较好的密码统计特性。

4.5 改进的 A5/1 混沌流密码算法的设计

通过前文对A5/1以及logistic混沌序列的分析，我们可以看到，A5/1流加密算法的密钥空间范围为$(2^{19}-1)(2^{22}-1)(2^{23}-1)\approx 2^{64}$，我们设计的logistic密钥空间范围为$2^{48}$。两者输出都具备良好的密码统计特性，为了增大密钥空间范围，提高加密算法的复杂度，我们将二者结合起来设计了一种改进型A5/1流加密算法。利用logistic混沌算法的密钥序列作为A5/1流密码发生器的初始密钥并参与运算，随着加密级别提高也可选用连续混沌系统或超混沌系统。其原理图如图4-8所示。

从原理图中可以看到，钟控单元在设计中有两个作用，一是控制 logistic 混沌发生器的序列密码输入与输出，另一个就是利用 3 个 LFSR 的位抽取信号对它们的移位方式进行钟控。当对 3 个 LFSR 初始化时，在 CP 作用下，移位寄存器工作；初始化结束即停止移位。Logistic 模型首先在 LFSR 初始的时候产生起始状态，另外在结束初始状态后参与密钥序列的运算中，因此该算法的非线性复杂度得到很大提高。因为密钥模块采用 logistic 混沌模型，代替传统的 LFSR 序列，利用混沌序列作为非线性组合序列函数一部分，与传统的 LFSR 序列相比，混沌序列周期长、复杂度高，对初态极端敏感，想恢复出混

沌序列是极其困难的，提高了密码的抗攻击性，另外通过改变迭代的初始值，可以得到大量的随机性能良好的混沌序列，解决了 LFSR 序列数量有限的问题。我们采用 LFSR 不规则钟控，并引入混沌序列，操作的基本步骤类似于 A5/1。LFSR 适合硬件的实现，能产生大周期的具有很好统计特性和伪随机性序列。不规则钟控和混沌序列把非线性因素引入到序列当中，提高其抗攻击能力。

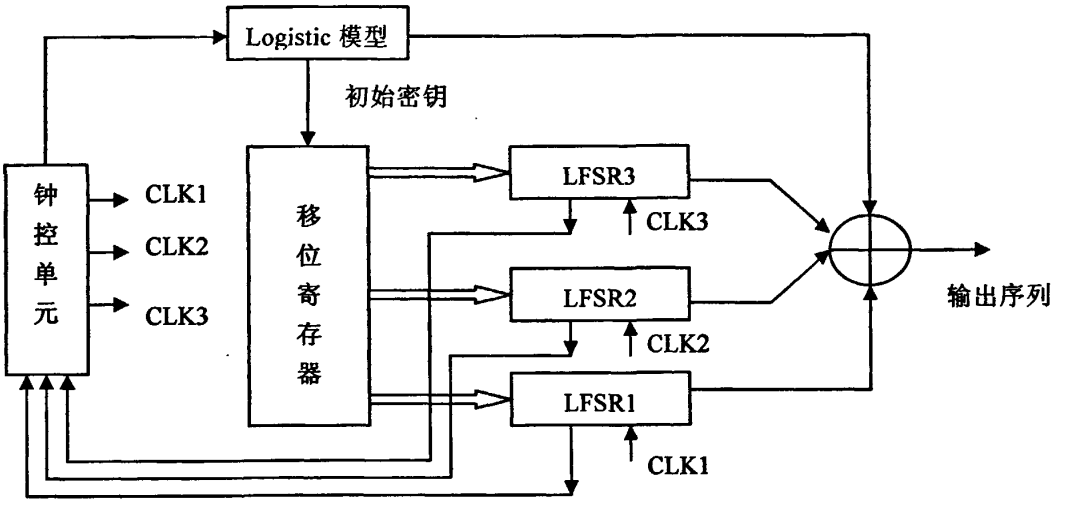


图 4-8 改进的 A5/1 混沌流密码算法原理图

Figure4-8 The schematic diagram of improved A5/1 chaotic stream cipher arithmetic

4.6 自同步序列密码的硬件实现

在研究了流密码同步以及混沌同步方案的理论基础上，我们分别进行了仿真实现，由于混沌同步技术受到计算机有限精度的影响以及严格同步性的实现困难，还不能够解决实时视频数据的加密传输，因此我们通过对同步流密码与自同步流密码加密结构进行比较和分析，发现自同步流密码在加密完并发送完数据后，在解密端解密时能够在很短的时间内与发送端的数据进行同步。这种传输方式非常适合于对实时性要求较高的视频加密传输中。以下我们研究设计了一个利用自同步这一新型流加密方式来进行实时视频传输的系统。

1、非线性组合函数的选择

非线性组合函数构造方法是根据 $n=4$ 或 $n=5$ 时的布尔函数 $f(x_1,x_2,\cdots\cdots,x_n)$，来构造出 $n\geq 6$ 时的布尔函数 $f(x_1,x_2,\cdots\cdots,x_n)$。

具体方法如下：

设 m,n 是两个不同的整数，$m>n$，m 表示所构造的非线性函数最高次数，n 代表基函数的次数。$I_{m,n}=\{I_{00},I_{01},I_{10},I_{11}\}$。$I_{m,n}$ 中包含 2^{m-n} 个函数，其中每个都满足严格的雪崩特性、具有相关免疫性、平衡性和代数式次数较高性质。

令 $x=(x_1,x_2,\cdots\cdots,x_n)$，$y=(y_1,y_2,\cdots\cdots,y_{m-n})$，那么构造的函数为 $f(y,x)=\oplus D_a(y)I_a(x)$。

其中 $a\in GF(2)^{m-n}$，$D_a(y)=(y_1\oplus a_1\oplus 1)(y_2\oplus a_2\oplus 1)\cdots\cdots(y_s\oplus a_s\oplus 1)$。

例如，取 $n=5$ 非线性函数，要求构造 $n=7$ 非线性函数，计算如下：

$$f_{00}(x_1,x_2,x_3,x_4,x_5)=x_1x_2x_3\oplus x_1x_4\oplus x_2\oplus x_5$$

$$f_{01}(x_1,x_2,x_3,x_4,x_5)=x_1x_2x_3\oplus x_1x_4\oplus x_2x_4\oplus x_2\oplus x_4\oplus x_5 \qquad (4\text{-}3)$$

$$f_{10}(x_1,x_2,x_3,x_4,x_5)=x_1x_2x_3\oplus x_1x_4\oplus x_1\oplus x_2\oplus x_5$$

$$f_{01}(x_1,x_2,x_3,x_4,x_5)=x_1x_2x_3\oplus x_1x_2x_4\oplus x_1x_3x_4\oplus x_1x_4\oplus x_2x_4\oplus x_3x_4\oplus x_1\oplus x_5$$

由此得到

$$\begin{aligned}f(y,x)&=\oplus D_a(y)I_a(x)\\&=(1\oplus y_1)(1\oplus y_2)f_{00}(x_1,x_2,x_3,x_4,x_5)\oplus(1\oplus y_1)y_2f_{01}(x_1,x_2,x_3,x_4,x_5)\\&\quad\oplus y_1(1\oplus y_2)f_{10}(x_1,x_2,x_3,x_4,x_5)\oplus y_1y_2f_{11}(x_1,x_2,x_3,x_4,x_5)\end{aligned} \qquad (4\text{-}4)$$

以 x_6,x_7 替代 y_1,y_2 得到

$$
\begin{aligned}
&f(x_1,x_2,x_3,x_4,x_5,x_6,x_7)\\
&=(1\oplus x_6)(1\oplus x_7)f_{00}(x_1,x_2,x_3,x_4,x_5)\oplus(1\oplus x_6)x_7f_{01}(x_1,x_2,x_3,x_4,x_5)\\
&\quad\oplus x_6(1\oplus x_7)f_{10}(x_1,x_2,x_3,x_4,x_5)\oplus x_6x_7f_{11}(x_1,x_2,x_3,x_4,x_5) \qquad (4\text{-}5)\\
&=x_2\oplus x_5\oplus x_1x_4\oplus x_1x_7\oplus x_4x_6\oplus x_1x_2x_3\oplus x_4x_6x_7\oplus x_2x_4x_6\oplus x_2x_6x_7\\
&\quad\oplus x_3x_4x_6x_7\oplus x_1x_2x_4x_6x_7\oplus x_1x_3x_4x_6x_7
\end{aligned}
$$

从实际计算可知，取$n=5$的非线性函数来构造$n=7$的同时满足几种良好特性的非线性函数计算起来已经很繁杂。

由于我们采取的是硬件加密方式，因此为降低芯片占用空间以及硬件实现难度，提高加密速度，我们选取了$f_{00}(x_1,x_2,x_3,x_4,x_5)=x_1x_2x_3\oplus x_1x_4\oplus x_2\oplus x_5$作为自同步流密码设计中非线性运算函数的基本运算单元。

2、加密系统原理

这一加密系统具有加密端和解密端，它在接收端和发送端都有密文反馈给系统当中的多位的移位寄存器。设计方案如下：由于非线性函数输入端口为 5 个，移位寄存器每次串行移动 16 位反馈密文，因此在发送端设定了一个 80 位的移位寄存器，并赋予其一个非秘密的初始状态。移位寄存器一次移动 16 位，其输入的 16 位值为明文序列与密钥序列异或后的密文序列。密钥序列发生器中包含 16 个相同的基本运算单元，每个单元将产生一位序列，即一共 16 位的输入与私有密钥 k 进行运算从而形成密钥序列 z，此密钥序列 z 与明文序列 p 进行异或形成所要传输的密文 c，在传输密文的同时把密文 c 一位一位的反馈回移位寄存器。而在接收端，将接收到的密文 c 反馈给 80 位的移位寄存器当中，这些密文与密钥进行运算得到与发送端相同的密钥序列 z，这个密钥再同刚才接收到的密文进行异或运算便可得到所发送的明文序列 p。

加解密端自同步模块如图 4-9 所示。该模块又由密钥序列发生器，密文反馈模块和缓存三部分组成。

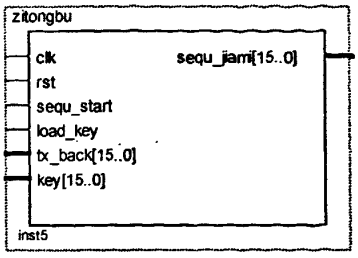


图 4-9 自同步模块

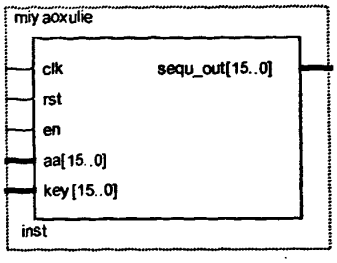


图 4-10 密钥序列模块

Figure4-9 self-synchronizing module

Figure4-10 key sequence module

（1）密钥序列发生器模块

密钥序列生成模块，即为图 4-10 中的 miyaoxulie 模块。此模块的主要作用是将反馈回来的密文数据与私有密钥进行一定的运算从而得到一条完整的密钥流序列。

该模块的主要程序如下：

```
process(clk,rst,en)
begin
    if rst='1' then sequ_out<=(others=>'0');
        elsif clk'event and clk='1' then
            if en='1'then
            sequ_out<=key xor aa;
            end if;
        end if;
end process;
```

在这个密钥流生成模块当中一共由 16 个小模块组成，他们为相同的数据运算单元，模块图如图 4-11 所示。该非线性函数的运算单元符合的多项式为：$f(x)=x_1x_2x_3\oplus x_2\oplus x_1x_4\oplus x_5$，其中 x_1,x_2,x_3,x_4,x_5 分别为该单元的 5 个输入端，而 $f(x)$ 则为该单元的逻辑输出端。这样经过此模块便可将输入端的 5 位数据变为一位而输出出来。这 5 位输入数据是移位寄存器的输出数据中选取的。在

这个密钥流生成模块中使用了 16 个这样的运算单元。而每一个运算单元的输入都是在移位寄存器的输出数据中选定的。从而在经过了这 16 个运算单元后将原有的 80 位数据变为了 16 位。这 16 位数据将作为 miyaoxulie 模块输入端口 aa[15..0]的输入值参与到密钥流序列的运算中去。

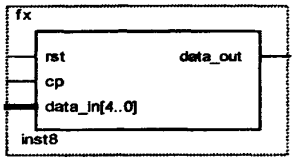


图 4-11 运算单元模块图

Figure4-11 arithmetic unit module

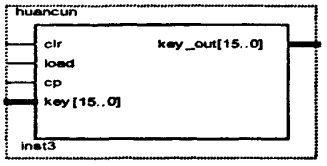


图 4-12 缓存模块

Figure4-12 cache module

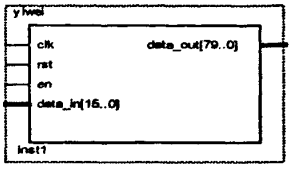


图 4-13 密文反馈模块

Figure4-13 ciphertext feeback module

该运算单元的主要程序如下：

```
process(cp,rst)
begin
   if rst='1' then   data_out<='0';
       elsif cp'event and cp='1' then
data_out<=(q(0) and q(1) and q(2)) xor (q(0) and q(3)) xor q(1) xor q(4);
       end if;
end process;
q<=data_in;
```

（2）缓存模块，即为图 4-12 中的 huancun 模块。其主要作用是将 16 位私有密钥进行缓存，实现串并转换输出。

该模块的主要程序如下：

```
process(cp,clr,load)
begin
   if clr='1' then   key_out<=(others=>'0');
       elsif cp'event and cp='1' then
```

```
            if load='1'then
               key_out<=key;
            end if;
        end if;
    end process;
```

将输出的密钥序列与所要加密的明文序列进行异或运算，从而得到我们所要传输的密文序列。

（3）最后一个模块便是体现了自同步思想的密文反馈模块，即为图 4-13 中的 yiwei 模块。在这里该模块为一个 80 位的移位寄存器，其主要作用是将密文生成模块生成的密文序列在发送的同时反馈给密钥流生成模块参与密钥序列的生成。移位寄存器每次移动 16 位密文。首先会为其赋予一个非秘密的初始状态，该寄存器通过 CP 来进行控制移位寄存器的移位，每过来一个 CP 脉冲便将密文数据向左移 16 位。其输入端为反馈回来的 16 位密文序列，输出将输入到密钥流发生器中经过非线性函数运算得到密文序列。

该模块主要程序如下：

```
process(clk,rst,en)
begin
   if rst='1' then data_out<=(others=>'0');-- q<=(others=>'0');
       elsif clk'event and clk='1' then
                if en='1'then
               data_out(15 downto 0)<=data_in;
               data_out(79 downto 16)<=data_out(63 downto 0);
               end if;
       end if;
end process;
```

（4）仿真结果：

CP，CP_jiemi 分别为加密端和解密端的时钟脉冲；m_input 为加密端的明文输入端口；m_output 为解密端的明文输出。

在此次仿真结果当中 CP，CP_jiemi 时钟脉冲的周期为 10ns，而加密端与解密端的密钥默认一致。m_input 为随机输入的一段明文序列。

从图 4-14 中可以很好的看出在经过了短暂的时间之后解密出的序列与输入的明文序列实现了自同步。

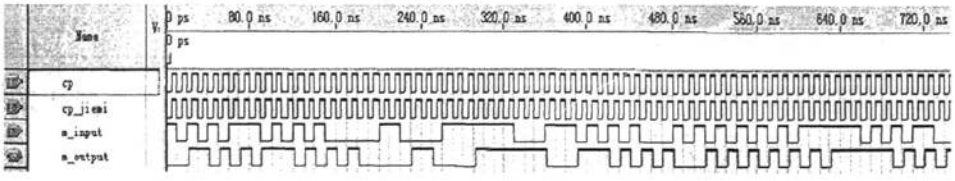


图 4-14 自同步流密码仿真图

Figure4-14 The simulation waveform of self-synchronizing stream cipher

4.7 本章小结

本章主要研究了混沌序列密码算法以及自同步流密码的设计与实现。由于混沌算法具有历性、混淆性、精确性以及对初始条件敏感的特性，与密码学中扰乱和扩散的密码特性较为接近，因此开展混沌密码学的研究是近年来密码学领域的一个研究重点。首先阐述了混沌学与密码理论的相关性，介绍了当前混沌密码学的主要设计模式，包括模拟混沌保密系统与数字化混沌密码设计。由于当前数字化混沌密码理论的研究主要都是基于软件实现，我们采用的硬件实现离散混沌密码设计具备较高的研究价值。然后，介绍了流密码算法中的 A5/1 算法，并详细介绍了我们设计的 logistic 离散混沌序列密码原理以及实现方案，利用 Matlab 下的 Dspbuilder 建模并实现了 VHDL 语言的转换，通过仿真验证实现了 logistic 离散混沌序列密码，并通过平衡性以及游程分析，验证了我们设计的密码序列具备较好的密码特性。但在实际应用中，由于混沌映射在计算机有限精度运算下会出现特性退化的现象，使其出现不

可避免的周期性。因此专门进行编程对 logistic 有限精度下的输出序列进行了周期测试，证明了其受计算机精度影响下的周期效应。为了增加我们设计的基于混沌的序列密码算法的复杂性，并减少因计算精度导致混沌特性退化的影响，我们最后设计了改进型 A5/1 混沌流密码算法，并通过加密测试验证了混沌对于初始值的敏感性。对于要求实时性较高的加密传输系统，根据自同步流密码的原理，我们又用硬件实现了一种自同步流密码方案，并通过仿真验证了同步结果。

第5章 视频加密传输系统的研究与实现

5.1 视频加密传输系统的整体构建方案

视频加密传输系统的整体框图如图 5-1 所示。整个系统由以下几个部分组成：CCD 摄像头实现视频模拟信号的采集；视频采集卡实现原始视频数据的编码和压缩；网络数据加密卡实现压缩后的视频信号的加密和网络传输；网络解密卡将接收到的加密信号经过解密运算得到编码后的视频信号；最后远程客户端软件实现信号解码、显示，从而完成加密传输，实现远程监控。

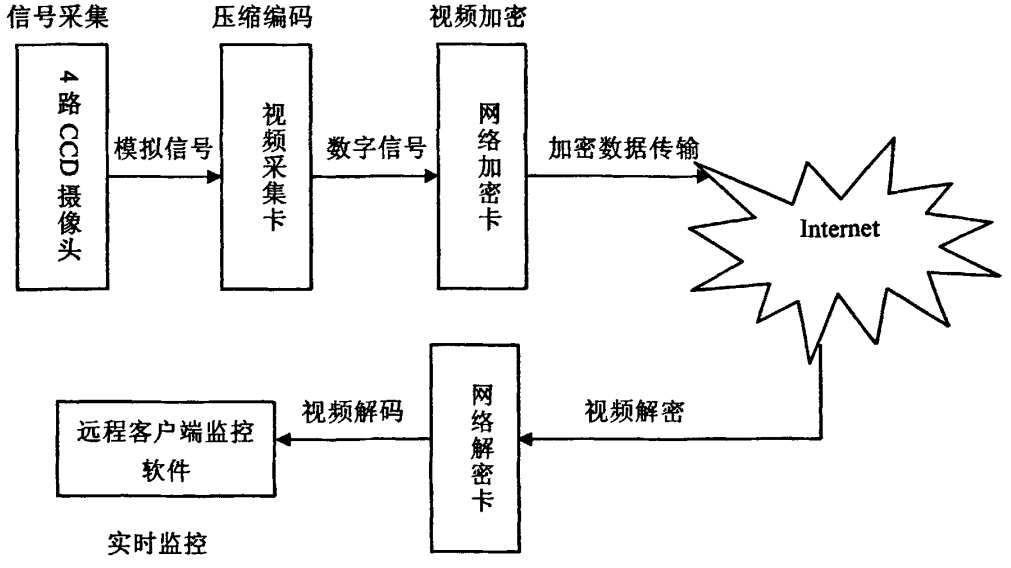


图 5-1 视频加密传输系统的整体框图

Figure5-1 The block diagram of video encryption transmission system

5.2 视频采集卡的选择

视频流数据的采集使用如下的硬件接口：硬件是基于三星公司(Samsung)的摄像头和杭州海康威视有限公司的音/视频 4 路硬压缩卡，视频编码是采用了最新的视频编码标准 H.264 来实现的，通过将摄像头采集来的原始 YUV 视

频流数据进行 H.264 编码，来进一步的压缩图像，以便于能够利用网络进行传输。

目前视频监控领域采用的视频采集卡有两种：一种是硬压缩卡，一种是软压缩卡。硬压缩卡采用的技术是使用板卡上的 DSP 进行视频图像的压缩编码，这样视频数据传到 PC 机上就已经完成了数据的压缩，而不耗费 CPU 的时间。软压缩卡刚好相反，板卡上只负责简单的视频数据的处理，真正完成压缩编码需要在 PC 机上进行，CPU 参与压缩编码，当压缩算法十分复杂的时候，耗费了 CPU 的大量时间。从视频传输的实时性以及二次开发的角度来进行分析，我们选择了海康威视的硬压缩卡实现视频信号的采集。

海康威视 DS-4004HC 是面向数字监控而推出的 4 路音/视频采集卡，采用的是高性能的视频压缩技术标准 H.264 及 OggVorbis(相当于 G.722)的音频编码标准，支持 4 路的 DCIF/2CIF/CIF/QCIF 实时编码压缩，同时还支持 2 路的 4CIF 实时编码压缩。提供 PAL 和 NTSC 两种视频制式，同时支持运动帧检测。它完全依靠硬件实现了视频及音频的实时编码，实现了动态码率、可控帧率、帧模式选择、动态图像质量控制，音频预览、视频丢失报警等功能，能独立调整各通道参数，性能稳定而且可靠。与 MPEG-I 产品相比，在保持同等图像质量的前提下，能大大节省存储空间、并非常适合宽带网或窄带网的传输，是新一代数字监控产品的最佳选择[79]。

我们的整个系统的实现利用了视频采集卡的 SDK 开发包，通过一些函数以及动态链接库实现了视频信号的采集并获取到编码压缩后的数据，将其传送到网络加密卡中，实现数据的加密和解密传输。

5.3 网络加密卡的设计

网络加密卡分为硬件和软件两部分。在硬件结构上是将 FPGA 作为主机硬件接口电路的主控制芯片，并在 FPGA 芯片中嵌入加/解密模块，利用混沌加密算法/自同步流加密算法对接收和发送数据帧中的明文数据/实时视频数

据进行加/解密。FPGA 设计网络加密卡的方法不会破坏 IP、路由等信息，而且适用于现有网络设备。软件部分在专用客户端程序中嵌入网络协议栈，采用独立于操作系统的方式对硬件的接口数据进行处理。

1、网络加密卡硬件电路设计

网络加密卡的 FPGA 加密芯片中主要包括控制模块、加密芯片模块和数据加/解密模块构成。其中加密芯片模块中的密钥序列发生器模块是基于混沌 A5/1 算法以及自同步流密码算法来分别设计的；数据加/解密模块实现密钥序列发生器与明文的流加密；控制模块负责控制整个加密过程的时序。除了网络加密芯片外，硬件电路板还包含配置芯片、网卡专用芯片、PCI 接口、内存芯片等接口电路设计。最终实现网络数据加密装置，使加密芯片通过 PCI 总线与设备进行数据交换，将普通数据进行加密传送。

2、网络加密卡软件程序设计

网络加密卡的软件设计过程，包括驱动程序的开发、网卡功能的实现、以及方便用户使用的显控界面的设计。

开发驱动程序采用的是 Driverstudio 开发工具。首先，通过 Driverstudio 开发的驱动程序的入口函数来查找并加载要访问的加密卡，然后它会自动锁定加密卡的资源只被开发者所用而不能被其它程序访问，随后用 Driverstudio 提供的读写函数控制主芯片的寄存器对 PCI 接口进行初始化、还包括主芯片寄存器、网卡 IP 地址、MAC 地址的设定。

网卡初始化成功后，启动接收线程，接收线程的应用程序采用事件驱动机制来和驱动程序通信，应用程序首先把信号量传递给驱动程序；当网卡接收到数据包后，会主动向 CPU 申请中断，然后启动驱动程序的中断函数，中断函数通过修改应用程序传递过来的信号量来启动用户接收线程，然后判断主芯片的写指针是否等于读指针，如果相等则比较收到数据帧的目的地址与加密卡的地址是否相同，如果相同则读取该数据包进行加密运算，否则丢弃。

网卡的发送过程是通过 TCP 协议进行文件的传输。传输过程有加密与非加密方式。无论是采用哪种，TCP 协议首先进行“三次”握手来建立连接，然后发送进程读取数据文件，根据数据帧帧格式中的文件类型来判断要传输是加密信息还是非加密信息，如果是非加密信息，数据经发送程序直接打包后发送；如果是加密信息，则进行硬件加密。发送端在等待 ACK 应答帧时，TCP 协议会采用定时器手段，在设定的时间内若没有收到应答帧，就取消发送任务，并显示发送失败，直到收到接收方 ACK 确认信号后，发送下一帧，依次握手发送全部数据信息。发送完成后要向接受方发送 FIN 结束帧表示发送结束。

网卡的接收进程与发送进程类似，建立“三次”握手连接后，接收进程便开启定时器，如果在设定时间内没有收到数据，就结束接收，并进行错误提示；如果正常接收数据，接收进程会读取数据帧头部信息，判断收到的是非加密还是加密文件，如果收到的是加密文件，接收程序先对加密芯片进行设置，再进行解密接收，同时发送 ACK 正确接收应答信号。在收到 FIN 信号结束数据发送后，发送 FINACK 确认并完成接收。

用户界面我们采用的是 VC 中的 MFC 技术。MFC 其实就是使用 C++类对 Windows API 进行了封装，它一方面屏蔽了 Windows 编程的复杂性，使得 Windows 应用程序的设计变得简单起来，另一方面仍然允许有经验的程序员使用 Windows 的底层开发高效的 Windows 应用程序。界面总共分为四个部分：串口的设置窗口、接收窗口、发送窗口以及发送选择窗口。发送选择窗口提供了发送普通信息和加密信息的选择，方便用户选择。另外考虑到界面的友好性，我们对所有需要发送的数据都重新创建了一个线程，这样就不会让用户在发大数据量时出现主界面不能使用或者死机的情况。发送和接收界面主要完成的是对线程处理完的数据或文件进行显示。

5.4 视频信号采集及播放器设计

5.4.1 视频信号采集二次开发

本设计中使用的视频采集卡是海康威视研发的具有 SDK（Software Development Kit）二次开发功能的 DS-4004HC 四路视频采集卡[79]。利用板卡所带的 SDK 开发包我们实现了视频信号采集和播放器的二次开发，进而实现了整个视频加密传输的客户端和服务器端的设计。

DS-4004HC 视频采集卡具备一个二次开发所需要的 SDK 程序包。在电脑上安装了采集卡的驱动以及程序开发包后，可以看到海康为用户二次开发提供了三种设计方案。在“SDK”文件夹下是二次开发需要的库函数和头文件，包含了板卡的 SDK，播放器 SDK，以及网络版的 SDK。“Doc”文件夹下是相关的开发文档说明，我们的二次开发参考了板卡和播放器的开发文档。“Src”文件夹下是海康威视做的 Demo 演示版本（采用 VC 语言编程），我们可以参考其提供的源代码进行二次开发，从而缩短开发周期。

板卡的 SDK 中提供了四个文件，“DS40xxSDK.dll”是运行板卡时调用的动态链接库；“DS40xxSDK.lib”是编译程序时告诉操作系统在运行时需要调用的 dll 文件；“HikVisionSdk.h”和“DataType.h”两个头文件用来对动态链接库外部函数以及变量的声明。在播放器的 SDK 中有三个文件，分别是“HikPlayM4.dll”、“HikPlayM4.lib”以及“HikPlayMpeg4.h”，它们的作用与板卡中的几个文件是一样的。

另外，我们要实现网络视频监控功能，播放器的二次开发包中他们对视频编码中的 DirectDraw 和 DirectShow 技术进行了封装，因此我们不必去再考虑压缩编码的 H.264 技术的实现，直接关注于对接收的视频数据进行处理即可。

参考了 Demo 版本视频采集卡本地视频信号显示的 SDK 程序，在海康提供的板卡开发说明中一共提供了近 60 个 API 函数，我们按需要对其进行了精

简，实现了视频信号的采集与显示。开发流程如图 5-2 所示，主要包括以下几个步骤：视频采集卡的初始化，实现本地预览模式，设置编码参数以及回调显示视频信号。

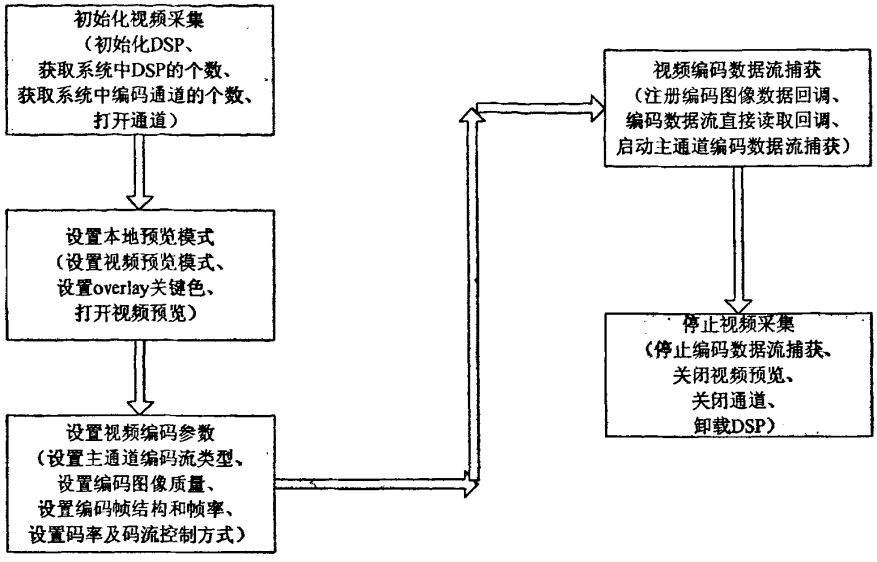


图 5-2 视频信号采集流程图

Figure5-2 The flow diagram of video signal acquisition

（一）视频采集卡的初始化 API

要开启视频采集，首先要完成的就是视频采集卡进行初始化设置，包括初始化 DSP 以及卸载 DSP，获取板卡信息等 API 函数。

1、初始化 DSP—InitDSPs

```
if (InitDSPs() <= 0)
{CString strMsg("打不开采集卡，请检查卡是否正确安装");}
```

函数说明：初始化系统中每一块板卡，应在应用软件程序启动时完成。如果返回值为 0 则表明初始化失败，没有找到相应的 DSP 模块。

2.卸载 DSP—DeInitDSPs

函数说明：关闭每一块板卡上的功能，在应用软件程序退出时调用。

3.获取板卡信息的 API 函数

（1）获取系统中 DSP 的个数—GetDspCount

函数说明：获取系统中所有板卡的 DSP 的个数。

(2)获取系统中编码通道的个数—GetEncodeChannelCount

调用：DLLEXPORT_API unsigned int __stdcall GetEncodeChannelCount();

函数说明：获取系统中所有编码卡的编码通道总个数。

（3）打开通道—ChannelOpen

程序调用：

```
for (i=0;i<GetTotalDSPs();i++)
{ChannelHandle[i]=ChannelOpen(i);//打开通道，获得 DSP 句柄}
```

函数说明：打开通道，获取编码通道的操作句柄。参数 ChannelNum：通道编号（从 0 开始）。

（4）关闭通道—ChannelClose

函数说明：关闭通道，释放相关资源

因为海康威视的视频采集卡分 4 路、8 路、16 路编码通道几种，因此先通过 GetDspcount()以及 GetEncodeChannelCount() 函数来获得所有板卡 DSP 个数以及编码通道数，接着再用 ChannelOpen()打开需要采集信号的通道，传递的参数是通道的编号。ChannelClose()函数用来关闭 ChannelOpen()打开的 DSP 编码通道，释放内存。

（二）视频采集卡本地预览模式 API

采集卡 SDK 开发包具备实现本地预览功能的 API，该功能采用的是 Overlay 预览模式将视频信号进行本地显示。Overlay 通常被称为重叠页面或者是覆盖层，是一种需要特定的硬件支持，直接可以将实时视频显示于主页面的技术。使用该方式进行预览可以提高预览的画面质量和减少 CPU 利用率。主要由以下 API 函数实现：

1、设置视频预览模式—SetPreviewOverlayMode

程序调用：

```
DLLEXPORT_API int_stdcall SetPreviewOverlayMode(BOOL bTrue);
int iOverlay_Mode=-1;//重叠
    iOverlay_Mode=SetPreviewOverlayMode(TRUE);
    TRACE("iOverLay_Mode=%4x\n",iOverlay_Mode);//调试
    if (iOverlay_Mode==0)
    {bOverLayMode=TRUE;}
```

传递参数：BOOL bTrue：是否设置 overlay 预览方式

函数返回值：0 表示显卡支持板卡的 overlay 预览方式；其他值表示显卡不支持 。

函数说明：当预览画面小于 704*576 时，需要调用该函数来启动 overlay 模式，如不设置则自动切换到 offscreen 模式进行预览显示，当预览画面大于 704*576 时，板卡自动切换到 overlay 模式。

2、设置 overlay 关键色 SetOverlayColorKey

函数：int_stdcall SetOverlayColorKey(COLORREF DestColorKey)

参数：DestColorKey：overlay 关键色参数（RGB（*，*，*））

函数说明：采集卡的显示范围小于 704*576 时，调用 SetPreview OverlayMode 打开 overlay 预览模式，需调用 SetOverlayColorKey 设置 overlay 关键色；当显示范围大于 704*576 时，板卡自动切换到 overlay 预览模式，默认颜色设置为 RGB（10，10，10），若要修改颜色设置，可以调用 SetOverlayColorKey 函数。但都需要将显示窗口的背景和关键色相一致，否则图像无法显示。 而且该函数调用要在 StartVideoPreview()前，否则将无效。

3.打开和关闭视频预览

（1）打开视频预览 StartVideoPreview

程序调用：BOOL bDdrawMode=FALSE;//DirectDraw 模式

```
if(bDdrawMode)
::StartVideoPreview(ChannelHandle[s],m_hWnd,&previewWnd,,FALSE,vdf
RGB16,25);
    else
    ::StartVideoPreview(ChannelHandle[s],m_hWnd,&previewWnd,FALSE,v
dfYUV422Planar,25);
    DisplayMode = VIDEO_SINGLE_CHANNEL;}
```

传递参数：HANDLE hChannelHandle：通道句柄 ；HWND WndHandle：显示窗口句柄 ；RECT *rect：显示窗口内的矩形区域 ；BOOLEAN bOverlay：是否启用 Overlay 预览模式；int VideoFormat：视频预览格式；int FrameRate：视频预览帧率（PAL：· 1-25，NTSC：1-30）。

函数说明：当调用 SetPreviewOverlayMode 开启视频预览以及 SetOverlayColorKey 设置好 overlay 预览颜色后，即调用 StartVideoPreview 函数启动预览，按需要选择预览的编码通道以及各个参数设置，然后就可以在画面中看到采集到的视频信号了。

（2）关闭视频预览 StopVideoPreview

程序调用：

```
for(int i = 0; i < GetTotalDSPs(); i++)
{StopVideoPreview(ChannelHandle[i]); }
```

传递参数： HANDLE hChannelHandle：通道句柄。

函数说明：当停止视频预览时，调用该函数关闭采集通道。

（三）视频采集卡视频编码参数配置 API

在编程指南中可以看到海康提供了双通道编码功能，即对一路视频图像可以进行两路视频编码，两路视频可以有不同的码流类型、不同分辨率、不同码率等。我们不需要实现双编码功能，因此对其进行了简化，只需要进行

主通道的编码参数设置即可。主要由以下 API 函数实现：

1、设置和获取主通道编码流类型—SetStreamType，GetStreamType

外部函数：

int_stdcall SetStreamType(HANDLE hChannelHandle, USHORT Type);

Int_stdcallGetStreamType(HANDLEhChannelHandle,USHORT *StreamType)

传递参数： HANDLE hChannelHandle（通道句柄）;

USHORT Type / *StreamType（编码类型/流类型）

函数说明：设置并获取主通道编码流类型。此函数需在启动编码前进行设置。在捕获采集之前，需要使用 SetStreamType()函数设置捕获采集通道的编码流类型。海康威视的视频采集卡支持的流类型有三种：视频流、音频流和音视频复合流。正常选择音视频复合流，但必须注意，此函数不能动态设置，必须在启动通道编码之前完成设置。设置成功后调用 GetStreamType()获取编码流格式。

2、设置编码图像质量—SetDefaultQuant

外部函数：

Int_stdcall SetDefaultQuant(HANDLE hChannelHandle, int IQuantVal,int PQuantVal, int BQuantVal)

传递参数：HANDLE hChannelHandle：通道句柄 ；int IQuantVal：I 帧量化系数；取值范围：12~30。int PQuantVal：P 帧量化系数；取值范围：12~30。int BQuantVal：B 帧量化系数；取值范围：12~30。

函数说明：设置图像量化系数，用于调整图像质量。量化系数越小图像质量越高；反之则越低。为了提高视频图像压缩算法，通常把几帧图像分为一组，但是为了防止图像的运动变化，分组的帧数不宜过多。H.264 采用的是 DCT（离散余弦变换）来实现压缩编码的，每组有三种类型的帧：分别是 I

帧、P 帧和 B 帧。I 帧是关键帧，只需要 I 帧就可以重构出图像画面；P 帧为前项预测帧，由 I 帧预测决定；B 帧为双向预测内插帧，它由 I 帧和 P 帧共同决定。由于 I 帧的信息量大，所以产生的图像质量也大，不易有过多的 I 帧，一般都是由 P 帧和 B 帧来进行预测插值。

SetDefaultQuant()函数用来设置编码图像质量，I 帧、P 帧、B 帧的量化系数取值范围都为 12-30，可以动态配置。量化系数影响着图像编码的质量和码速率的参数，量化系数越高，编码图像的质量就越低，所占用带宽就越小，相反，需要高质量画面就需要减小量化系数，默认设置是 18、18、23，可根据所需图像质量的要求以及网络环境动态修改。

3、设置编码帧结构和帧率—SetIBPMode

程序调用：

```
for (int i=0;i<4;i++)
{   ::SetIBPMode(ChannelHandle[i],100,2,0,15);// 帧结构和帧率
    SetupBitrateControl(ChannelHandle[i],150000);// 码流最大比特率
    SetBitrateControlMode(ChannelHandle[i],brVBR);}//码流控制方式
```

传递参数：HANDLE hChannelHandle：通道句柄 ；int KeyFrame Intervals：关键帧间隔；int BFrames：B 帧数量.；int PFrames：P 帧数量；int FrameRate：帧率。

函数说明：设置编码帧结构和帧率。

关键帧为视频流在压缩时采用帧内压缩的图像帧，优点是图像质量高清晰，但是数据量大，主要用于作为编码时的原始参考帧。关键帧间隔为连续帧间编码帧的个数，而 H.264 为有损压缩，所以设置关键帧间隔将影响到图像的质量和大小。其取值范围是 1-400，默认为 100。

帧率为每秒钟传输的图片帧数，取值范围取决于视频制式，国际上采用两种视频制式：PAL 制式和 NTSC 制式。PAL 为每秒 1-25 帧，NTSC 为每秒

1-30 帧，中国采用的是 PAL 制式，即帧率的范围为每秒 1-25 帧。

若带宽足够，而画面质量不够清晰，可以减小关键帧间隔和增大帧率。但若网络条件和差或需要在窄带下传输，就应该适当提高关键帧间隔和减小帧率。合理调整其参数值将影响视频传输的效果。

4、设置码率及码流控制方式

（1）设置码流最大比特率 SetupBitrateControl

函数：int_stdcall SetupBitrateControl(HANDLE hChannelHandle, ULONG MaxBps)

传递参数：HANDLE hChannelHandle：通道句柄 ；ULONG Maxbps：最大比特率。

函数说明：设置编码的最大比特率，取值要求在 10000 以上。

当设置为 0 时码流控制无效，设置为某一最大比特率时，当编码码流超过该值时，DSP 会自动调整编码参数来保证不超过最大比特率，当编码码流低于最大比特率时，DSP 不进行干涉。

（2）设置码流控制方式 SetBitrateControlMode

函数：int_stdcall SetBitrateControlMode(HANDLE hChannelHandle,

BitrateControlType_t brc)

传递参数：HANDLE hChannelHandle：通道句柄 ；BitrateControlType_t brc：码流控制方式 。

函数说明：设置编码码流控制方式。

码流控制方式有变码率（brVBR）和恒定码率（brCBR）两种方式。当设置为变码率（brVBR）时，最大比特率将作为编码码流上限，由 DSP 在码流上限范围内自动控制码率，一般会自动降低为最低的码率（由前面设定的编码图像质量参数和关键帧间隔来决定），这种方式能最大程度地降低传输带宽和所需存储空间，但难以估算存储容量；当设置为定码率（brCBR）时，则以

最大比特率恒定输出码流，存储容量可根据设定码率的大小进行估算，但会浪费存储空间，实际中可按需要来调整。

（四）视频采集卡编码数据流捕获 API

初始化和配置好视频采集卡后，接下来就是进行捕获采集到的视频数据。板卡 SDK 开发包提供三种方式捕获编码后的 H.264 格式的视频数据，我们采用的是直接读取方式进行捕获的。完成视频数据捕获功能的主要有以下几个函数：

1、注册编码图像数据回调函数—RegisterStreamDirectReadCallback

程序调用：

RegisterStreamDirectReadCallback(::StreamDirectReadCallback,this); // 注册读取视频流的回调函数

传递参数：

STREAM_DIRECT_READ_CALLBACK StreamDirectReadCallback：编码数据流直接读取回调函数 ；void* Context：设备上下文 。

函数说明：当启动数据捕获后，StreamDirectReadCallback 会提供数据流的地址、长度、帧类型等，供用户程序直接处理。

编码数据流直接读取回调函数的参数如下：

Typedef int (*STREAM_DIRECT_READ_CALLBACK)(ULONG channel Number, void *DataBuf,DWORD Length,int FrameType,void *context) ：ULONG channelNumber：通道号；void* DataBuf：缓冲区地址；DWORD Length：缓冲区长度；int FrameType：缓冲区数据帧类型；void* context：设备上下文。

接下来通过需要重写注册的回调函数，在回调函数里可以获得捕获到的压缩视频图像。得到数据后，把数据拷贝到一个缓冲区去，不在回调函数里面对数据进行任何处理。

2、启动主通道编码数据流捕获—StartVideoCapture

程序调用：

```
for (int i=0;i<GetTotalDSPs();i++)
{StartVideoCapture(ChannelHandle[i]); }
```

传递参数：HANDLE hChannelHandle：通道句柄

函数说明：启动主通道编码数据流捕获。最后一步需要启动视频捕获，使用 StartVideoCapture()函数启动编码数据捕获。动态链接库中每采集到图像数据，都会调用回调函数，这个时候就可以从回调函数中获取到数据进行处理。当捕获数据结束时，调用 StopVideoCapture()函数来停止编码数据流捕获。

5.4.2 H.264 视频播放器二次开发

为了能够将我们采集到的 H.264 格式的视频图像通过网络进行实时监控，需要开发一个支持 H.264 格式的视频播放器。由于视频播放器的设计需要用到 DirectDraw 和 DirectShow 技术，这将涉及到显卡驱动的操作，具有较高的难度。为了缩短开发周期和减小开发难度，我们参照海康威视提供的 H.264 播放器 Demo 进行了二次开发。

海康威视播放器显示部分采用了 DirectDraw 技术，根据显卡是否支持缩放功能，可以使用两种方式进行显示：采用 Overlay 画面显示以及使用 offscreen 画面并使用 Blt 技术到主画面。海康威视的播放器 Demo 中提供了 97 个 API 接口函数，我们按需要进行了精简，开发播放器的主要流程有：初始化播放器，设置播放器参数，调用解码回调以及解码数据进行显示。

（一）播放器初始化 API

1、放器模式—Hik_PlayM4_SetStreamOpenMode

程序调用：

```
for (i=0;i<4;i++)
{Hik_PlayM4_SetStreamOpenMode(PORT+i,STREAME_REALTIME);//
```

设置播放模式}

传递参数：nMode：STREAME_REALTIME 实时模式（默认），STREAME_FILE 文件模式。其中，STREAME_REALTIME 是默认的实时模式，适合播放网络实时数据，尤其是要求带宽不是很宽的情况，这样播放器会对放在缓冲区的编码数据立刻解码，达到实时效果。STREAME_FILE 是文件模式，是将数据流转换为文件形式再播放，可以实现更清晰的播放效果。

函数说明：在打开播放器之前，必须先使用 Hik_PlayM4_SetStream openMode()设置播放的模式。

2、打开和关闭播放器数据流接口—Hik_PlayM4_OpenStream、Hik_PlayM4_CloseStream

程序调用：

```
for (int i=0;i<4;i++)
{if (m_bOpen[i])//打开就得关闭，重新打开
    {Hik_PlayM4_CloseStream(PORT+i);
     m_bOpen[i]=FALSE;  }
if (!Hik_PlayM4_OpenStream(PORT+i, glb_headerData,m_nHeadSize,
200000))//播放文件头}
```

传递参数：nPort：播放的是第几通道；pFileHeadBuf：存放文件头数据；nSize：文件头长度；nBufPoolSize：设置播放器中存放数据流的缓冲区大小。

函数说明：打开和关闭数据流接口；缓冲区范围是 50K—100M。

（二）播放器参数设置 API

1、播放文件的文件头长度—Hik_PlayM4_GetFileHeadLength()

程序调用：

```
for (i=0;i<4;i++)
{Hik_PlayM4_SetStreamOpenMode(PORT+i,STREAME_REALTIME);//
设置播放模式}
```

m_nHeadSize=Hik_PlayM4_GetFileHeadLength();//得到文件长度函数

函数说明：主要应用在 H.264 流播放器的 STREAME_FILE 模式下，该函数得到要播放的文件的文件头长度，将其作为 Hik_PlayM4_OpenStream()的输入参数。

2、设置缓冲区数据解码回调函数—Hik_PlayM4_SetSourceBufCallBack

函数：BOOLHik_PlayM4_SetSourceBufCallBack

(LONG nPort, DWORD nThreShold, void (CALLBACK SourceBufCallBack) (long nPort, DWORD nBufSize, DWORD dwUser, void *pResvered), DWORD dwUser, void *pReserved);

传递参数：nThreShold：阈值；SourceBufCallBack：回调函数指针；dwUser：用户数据；pReserved：保留数据。

回调函数：

Void CALLBACK SourceBufCallBack (long nPort, DWORD nBufSize, DWORD dwUser, void*pContext)

传递参数：nPort 播放器通道号；nBufSize 缓冲区中剩余数据；dwUser 用户数据；void*pContext 设备上下文指针。

函数说明：该函数设置源缓冲区阈值以及缓冲区内剩余数据小于等于阈值时的回调函数指针。

解码工作是由动态链接库完成，客户端接收并往解码缓冲区存放编码数据，然后绑定解码回调函数，设置一个阈值。当缓冲区数据剩余的数据小于等于阈值时，调用回调函数。在回调函数中可以唤醒读取文件和输入编码数据的线程。软件解码 CPU 的使用率很高，但是设置数据解码回调函数，并且通过一个通知事件，在解码完成回调函数中通知读取文件数据的方式可解决 CPU 占用率高的问题。

（三）解码数据显示 API

1、绑定播放器窗口—Hik_PlayM4_Play

程序调用：

```
Hik_PlayM4_Play(PORT,GetDlgItem(IDC_SHOW0)->m_hWnd);
Hik_PlayM4_Play(PORT+1,GetDlgItem(IDC_SHOW1)->m_hWnd);
Hik_PlayM4_Play(PORT+2,GetDlgItem(IDC_SHOW2)->m_hWnd);
Hik_PlayM4_Play(PORT+3,GetDlgItem(IDC_SHOW3)->m_hWnd);
```

传递参数：nPort 通道号；hWnd 播放视频的窗口句柄。

函数说明：当播放开始时，使用 Hik_PlayM4_Play()来绑定一个窗口句柄，播放视频画面大小将根据 hWnd 窗口调整，要全屏显示，只要把 hWnd 窗口放大到全屏。为避免画面重叠，不同的通道选择的窗口句柄应该不一样。

2、视频编码数据输入解码缓冲区—Hik_PlayM4_InputData

```
switch(pBuff->udpvideo.channel)
    {case 0x00:
    if(!Hik_PlayM4_InputData(PORT,pBuff->udpvideo.videodata,len_Data))
        {   strMsg.Empty();
            strMsg.Format("通道 0 关闭，视频流输入不进");
            Log_Data::SendEvent(strMsg);
        }
        if (m_bSaveFile[0])
        {_write(gFileHandle[0],pBuff->udpvideo.videodata,len_Data);// 存 储
视频文件  }
        break;
    case 0x01:
    if(!Hik_PlayM4_InputData(PORT+1,pBuff>udpvideo.videodata,len_Data
))
```

```
            {       strMsg.Empty();
                strMsg.Format("通道 1 关闭，视频流输入不进");
                Log_Data::SendEvent(strMsg);
            }
            if (m_bSaveFile[1])
            {_write(gFileHandle[1],pBuff->udpvideo.videodata,len_Data);// 存 储
视频文件  }
            break;
        case 0x02:
        if(!Hik_PlayM4_InputData(PORT+2,pBuff>udpvideo.videodata,len_Data
))
            {  strMsg.Empty();
                strMsg.Format("通道 2 关闭，视频流输入不进");
                Log_Data::SendEvent(strMsg);
            }
            if (m_bSaveFile[2])
            {_write(gFileHandle[2],pBuff->udpvideo.videodata,len_Data);// 存 储
视频文件  }
            break;
        case 0x03:
        if(!Hik_PlayM4_InputData(PORT+3,pBuff>udpvideo.videodata,len_Data
))
            {  strMsg.Empty();
                strMsg.Format("通道 3 关闭，视频流输入不进");
                Log_Data::SendEvent(strMsg);
```

```
}
if (m_bSaveFile[3])
{_write(gFileHandle[3],pBuff->udpvideo.videodata,len_Data);// 存 储
```

视频文件 }break;

传递参数：nPort 通道号；pBuf 解码缓冲区地址；nSize 缓冲区大小

函数说明：输入从卡上得到的流数据；打开流之后才能输入数据。TURE，表示已经输入数据。FALSE 表示失败，数据没有输入。Hik_PlayM4_ InputData() 把来自于网络上的或者是存放在磁盘文件中的视频编码数据放进解码缓冲区，调用动态链接库的解码线程进行解码。支持的文件格式是来自海康视频采集卡硬编码之后的 H.264 的视频格式。

5.4.3 视频信号监控及视频播放器的测试

通过对视频信号采集以及视频播放器设计的 API 接口函数的分析，我们设计了适合我们整个系统构建的简化版应用程序。如图 5-3 所示，通道 0 接入了视频摄像头，为采集到的视频信号，其他三个通道置空，默认显示“无视频信号”。

在远程服务器端设计了既可以播放网络解码视频数据又可以读取本地磁盘文件的 H.264 视频播放器测试软件，图 5-4 为 H.264 视频播放器测试软件，该测试软件的右侧为四路视频通道监控以及回放窗口，在左侧可以按保存的路径以及存储时间来检索本地磁盘保存的视频文件。

通过两个测试软件的设计我们成功实现了对视频信号采集以及远端存储播放的二次开发。

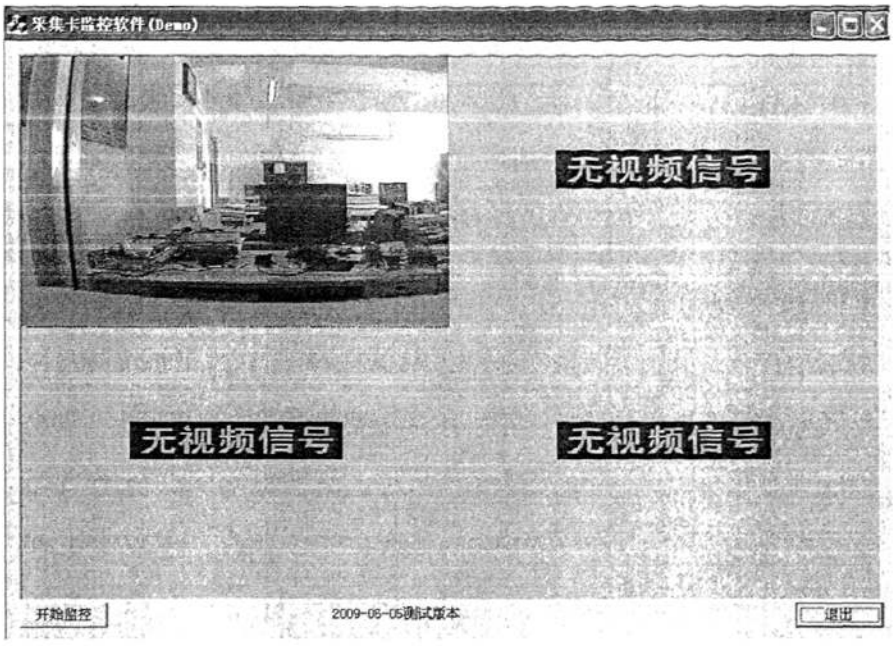


图 5-3 视频信号监控测试软件

Figure5-3 The testing software of video signal monitoring

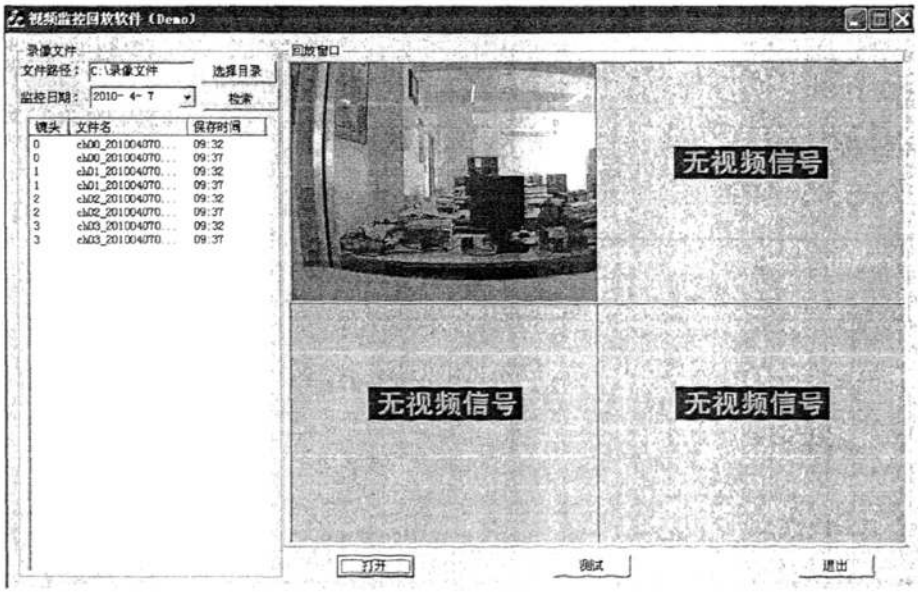


图 5-4 H.264 视频播放器测试软件

Figure5-4 The testing software of H.264 video player

5.5 视频加密传输系统的实现

整个视频加密传输系统包括客户端和服务器端两个部分组成。通过前面对网络加密卡的介绍，视频采集卡的二次开发以及 H.264 播放器的设计，我们将其综合起来，实现了视频加密传输系统的开发。

5.5.1 服务器端软件设计

服务器端软件应用程序要完成的功能有对视频信号的采集以及加密传输，另外还要进行本地视频监控。在服务器端软件设计中利用了上一节对视频采集卡的二次开发实现了本地视频信号监控，并且综合了网络传输协议，从而实现了视频信号采集，数据发送以及数据接收这三个线程的共同作用，最后开发了服务器端应用程序。

视频采集线程和发送线程之间有一个缓冲区，采集线程把从视频采集卡采集到的编码数据输入到缓冲区内，通知发送线程有需要发送的新数据。发送线程从缓冲区中提取数据，通过动态链接库编写的协议栈对编码数据进行打包，然后通知设备驱动程序，设备驱动程序读取数据后，直接通过总线传送到给网络加密卡，最后网络加密卡对数据进行加密处理，然后通过以太网芯片经网络传输。接收线程采用 UDP 方式对视频数据封装，发送视频数据之前，首先向客户端发送一个请求命令，确定客户端是否空闲，若服务器收到客户端的应答，就可以在接收线程中开启采集功能，进行数据采集以及传输。若服务器无响应，超时等待数秒进行重新发送，直到五次发送失败，认为对方不在网络或网络繁忙等。

5.5.2 客户端软件设计

客户端软件应用程序要完成的功能有接收网络传来的视频数据并进行实时监控显示，能够对各通道视频信号进行录像保存，实现文件检索功能。客户端的软件应用程序设计利用了上一节中视频播放器的二次开发来实现。客

户端有三个工作线程：接收线程，解码线程和文件存储线程，三个线程共同处理视频数据，另外主线程有一个 1s 的定时器，用来定时录制保存文件。

首先三个线程之间共享着一个公有缓冲区，接收线程将接收网络中的数据并进行数据分析，当发现是握手命令时，需要回复服务器端一个应答。当收到是视频数据时，把端口号，帧格式，帧数据以及长度等信息保存到缓冲区链表中，然后通知解码线程从缓冲区内里读取视频编码数据，再通过网络播放器进行解码显示。若这时定时器发现此时需要保存文件，就启动文件存储线程访问本地磁盘，创建并存储视频文件。

5.5.3 视频加密传输系统的测试

通过以上我们对整个视频加密传输系统的服务器端以及客户端的设计，我们搭建了网络视频加密传输系统，图所示即为服务器端视频传输与客户端实时监控显示的画面。从图 5-5 中可以看到经过短暂的时间，接收端即与发送端实现了视频信号的同步监控。

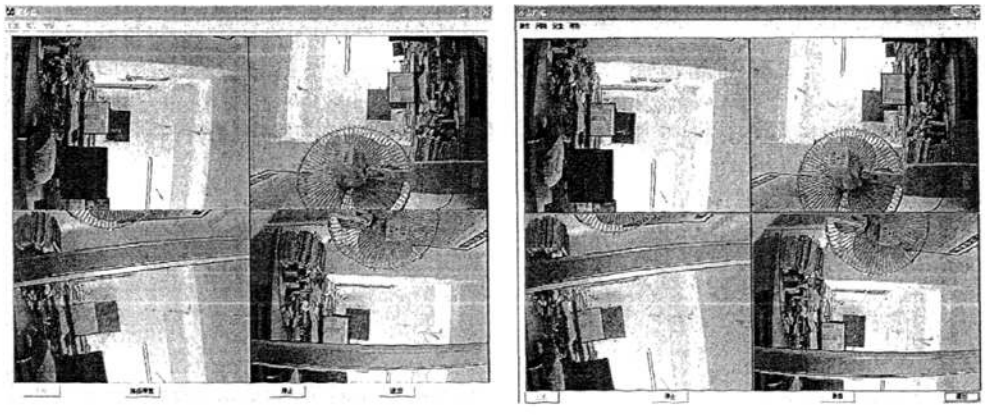


图 5-5 视频加密传输系统服务器与客户端监控画面

Figure5-5 The monitoring pictures of the server and client-side of video encryption transmission system

我们的传输实验中，视频信号的参数设置为：采用 overlay 标准视频预览模式，16 位 RGB 视频压缩格式，视频预览模式为 PAL，码流控制方式为变码率（brVBR），最大比特率设置为 150000bps，I、P、B 帧的默认量化系数值分

别为为 18，18，23；I 帧关键帧间隔为 400，视频编码帧率为 15 帧。按照实际要求，用户可以按需采用适合的视频参数进行设置。我们组成了十台微机的网络系统，分别按照用户使用说明进行硬件网络加密卡的安装和驱动程序的安装，然后进行驱动程序和应用程序的导入，系统建立后，分别按照数据、文件、视频信息三种形式，进行加密和非加密数据传输。针对于视频加密传输系统，我们进行了 24 小时不间断监控以及存储测试，经测试系统能够稳定工作。

5.6 本章小结

本章实现了视频加密传输系统的整体构建。首先介绍了视频加密传输系统的原理框图，然后介绍了视频采集卡的选择以及网络加密卡的硬件与软件设计，接下来参照海康的 Demo 应用软件，我们利用其 SDK 二次开发包设计实现了视频加密传输系统的视频采集和播放部分的二次开发。详细介绍了我们利用的 SDK 开发包中的 API 函数以及各部分设计过程，最后实现了针对板卡的 SDK 二次开发以及 H264 格式视频文件的视频解码播放器的二次开发，对开发过程进行了详细的说明，并验证了其采集与显示结果。最后按照项目需要，我们设计实现了视频加密传输系统的服务器端与客户端软件，并通过测试进行了验证。

结 论

本文研究了基于混沌算法的网络数据加密传输系统，并利用自同步流加密方法实现了实时视频信号的加密传输以及监控显示。首先从传统密码学角度入手，介绍了流密码算法的原理及应用，阐述了混沌算法理论，探讨了混沌方程在确定性方程下的不确定输出以及对初始条件极其敏感的特性，运用数值分析方法模拟了一维logistic混沌方程以及三维的Lorenz混沌方程的分叉图、吸引子图、Lyapunov指数谱图以及Kolmogorov熵谱图。然后通过simulink的数学建模方法，对模拟混沌同步保密通信中的混沌遮掩以及混沌键控方法进行了研究和仿真实现。在阅读了大量混沌密码相关研究论文的基础下，分析了当前混沌在保密通信中的应用方向和研究领域，提出了数字化混沌密码算法设计思想。利用 FPGA 与 Matlab Dspbuilder 结合设计了离散logstic混沌序列密钥生成器；通过对计算机有限精度下离散混沌周期性的探讨，我们进一步设计出改进型A5/1算法的混沌序列密钥生成方法，实现了网络数据加密卡中的混沌加密核的设计。针对于实时视频流的加密传输，我们又研究了一种自同步流密码方案，进而实现了集网络视频采集、加密传输、视频解码显示、以及监控保存等多功能为一体的实时视频流加密传输系统。本文的研究工作得到的结论主要有：

对传统密码学算法进行了总结与归纳，着重研究了序列密码算法的原理与实现方法；深入学习了混沌算法理论，对logistic以及Lorenz混沌系统进行了数值仿真，更加形象而直观的分析了混沌方程的各种特性，为混沌加密算法设计打下了理论基础。

综合分析了国内外对混沌保密通信的研究领域，对混沌理论中模拟混沌同步保密通信进行了探讨分析。通过 Simulink 图形化仿真，实现了混沌遮掩以及混沌键控等相关保密通信方案。对另一个研究领域—数字化混沌密码的设计提出了新的理念。对于计算机有限精度下，离散logistic方程出现周期特性进行了实验仿真计算，最终采取了离散 logistic 序列密码与传统的 A5/1 流密码相结合的方式，用

硬件语言实现了数字化混沌密码，并将其下载到 FPGA 中作为核心算法加密核，实现了网络数据加密传输功能。

对于视频数据流的加密传输来说，实时性很重要，因此视频信号发送端与接收端的同步性问题就变得尤为突出，采用混沌加密算法可以实现视频文件的传输，但当实时传输视频信号进行监控的时候，就可能会出现接收端与发送端视频加解密不同步的问题。因此设计了一种自同步流密码加密解决方案，利用视频采集卡的 SDK 进行了二次开发，成功实现了一个集视频采集、加密传输、视频监控、以及视频存储等多功能为一体的实时视频流加密传输系统。

混沌理论作为近年来信息安全领域的新生力量，正在受到越来越广泛的研究。我们的研究工作创新性的将混沌算法进行了硬件实现，并将其用到了实际应用中来，对于今后的研究工作具有指导性意义。但是从研究过程中可以发现，由于计算机有限精度效应等影响，数字化混沌算法在应用中已经失去了一些混沌特性，这种退化将直接影响到混沌算法在实际应用中的推广使用，而且当前针对离散化混沌序列密码的攻击方法以及加密算法的可靠性还有待进一步深入的研究。但我们有理由相信，基于混沌算法的加密方法定将成为信息安全应用中一颗璀璨的明珠。

参考文献

[1] 王超.基于网络的 H.264 视频监控系统的研究与实现[D].硕士论文,上海交通大学,2007.

[2] 胡向东,魏琴芳.应用密码学教程[M].北京:电子工业出版社,2005.

[3] 黄润生,黄浩.混沌及其应用(第二版)[M].武汉大学出版社,2005,12.

[4] R.Brown, L.O.Chua.Clarifying chaos: Examples and counterexamples [J].Int.J. Bifurcation and Chaos, 1996, 6(2):219-249.

[5] Dent, A.W., C.J.Mitchell.User's Guide to Cryptography and Standards [M]. Norwood, MA: Artech House Publishers, 2004.

[6] Davis R.The data encryption standard in perspective [J].IEEE Communications Magazine.1978, 16(6):5-9.

[7] Tianjie C, Xianping M.Collusion Attack on a Server-Aided Unbalanced RSA Key Generation Protocol [A], International Conference on Communication Techno -logy, ICCT'06[C].Guilin, China, Nov.2006:1-3.

[8] 时映鹏.基于 FPGA 技术的网络加密卡[D].硕士论文,黑龙江大学,2007.

[9] Oppliger R.Internet and Intranet Securit, 2nd edition [M]. Norwood, MA: Artech House Publishers, 2002.

[10] 廉士国,孙金生,王执铨.视频加密算法及其发展现状[J].信息与控制,2004,33(5): 560-566.

[11] Douglas R.Stinson.Cryptography: Theory and Practice [M].USA: CRC Press, 1995.

[12] C.E.Shannon.Communication theory of secrecy systems [J].Bell System Technical Journal.1949, 28(4):656-715.

[13] 冯登国,裴定一.密码学导引[M].北京:科学出版社,1999:56-59.

[14] A.Menezes, P.van Oorschot, S.Vanstone.Handbook of Applied Cryptography [M].

Boca Raton: CRC Press, 1997:191-212.

[15] N.Proctor.A Self-Synchronizing Cascaded ciper system with dynamic control of error propagation [J].Advanced in Cryptology—CRYPTO.1985, 84(3):174- 190.

[16] U.M.Maurer.New Approaches to the design of selfsynchronizing Stream Cihers [J].Advances in Cryptology-EUROCRYPT.1991, 91(3):458-471.

[17] 杨义先,钮心忻.应用密码学[M].北京:北京邮电大学出版社.2005:97-98.

[18] Lu Xiao,Stafford Tavares,Amr Youssef,Guang Gong.Improved Cascade Stream Ciphers Using Feedback[C].Conference on Electrical and Computer Engineering. 2005:104-108.

[19] Edward N.Lorenz.Deterministic non-periodic flow [J].Atmospheric Sciences, 1963, 20:130-141.

[20] Edward N.Lorenz.The Essence of Chaos [M].USA: University of Washington Press, 1993.

[21] Edward N.Lorenz.The predictability of hydrodynamic flow [J].Trans.NY. Academy of Sciences Series II.1963 (25):409-432.

[22] T.Y.Li, J.A.Yorke.Period Three Implies Chaos [J].The American Mathematical Monthly.1975, 82(10):985-992.

[23] 吴祥兴,陈忠等.混沌学导论[M].上海:上海科学技术文献出版社,2001:16-17.

[24] 卢侃,孙建华,欧阳容白,黄有来.混沌动力学[M].上海:上海翻译出版社.1990: 48-49.

[25] 王兴元.复杂非线性系统中的混沌[M].北京：电子工业出版社.2003(6):20-39.

[26] 刘娟.基于混沌系统的保密通信研究[J].电信交换.2009(3):1-4.

[27] R.L.Devaney.An Introduction to Chaotic Dynamical Systems [M].New York: Addison Wesley Massachusetts, 1989:8-49.

[28] 丁丽娜.基于 FPGA 的混沌加密芯片的应用研究[D].硕士论文,黑龙江大学, 2008.

[29] 于娜.混沌同步技术在保密通信中的研究与应用[D].硕士论文,黑龙江大学,

2007.

[30] L.M.Pecora,T.L.Carroll.Synchronization in Chaotic Systems[J].Physical Review Letters.1990, 64(8):821-824.

[31] K M Short.Steps toward Unmasking Secure Communication[J].Int.J.Bifurcation and Chaos.1994, 4(4):959-977.

[32] T Yang.Recovery of Digital Signals from Chaotic Switching [J].Int J.Circuit Theory Appl.1995, 23(6):611-615.

[33] Dedlieu H, Kennedy M P, Hasler M.Chaos Shift Keying: Modulation and Demodulation of Chaotic Carrier Using Synchronization Chua's Circuits [J]. IEEE Trans.Circuit&Syst.(II).1993,40(10):634-642.

[34] Philip Downes E-Systems, Inc.Secure Communication Using Chaostic Synchronization [J].1993, SPIE (2038):227-233.

[35] Milanovic V., Zaghoul M.E.Improved Masking Algorithm for Chaotic Communications Systems [J].Electronics Letters.1996, 32(1):11-12.

[36] 赵耿,方锦清.混沌通信分类及其保密通信的研究[J].自然杂志.2003, 25(1): 21-30.

[37] Dedlieu H, Kennedy M P, Hasler M.Chaos Shift Keying: Modulation and Demodulation of Chaotic Carrier Using Synchronization Chua's Circuits [J].IEEE Trans.Circuit&Syst.(II).1993,40(10):634-642.

[38] Kolumbán G.,Vizvari G.K.,SchwarzW.,Abel A. Differential chaos shift keying:A robust coding for chaos communication[C].Proc.International Workshop on Nonlinear Dynamics of Electronic Systems, Seville, Spain, 1996:87-92.

[39] Kolumbán G.Theoretical noise performance of correlator-based chaotic communications schemes [J].IEEE Transactions on Circuits and Systems Part I, 2000, 47(12):1692-1701.

[40] Kolumbán G.Performance evaluation of chaotic communications systems: Determination of low-pass equivalent mode[C].Proc. International Workshop on

Nonlinear Dynamics of Electronic Systems (NDES'98),Budapest, Hungary, 1998:41-51.

[41] Dedieu H., Kennedy M.P., Hasler M.Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuit [J].IEEE Transactions on Circuits and Systems Part II, 1993, 40(10): 634-642.

[42] Parlitz U., Chua L.O., Kocarev L., Halle K.S., Shang A.Transmission of digital signals by chaotic synchronization [J].International Journal of Bifurcation and Chaos, 1992(2):973-977.

[43] Kolumbán G., Kennedy M.P., Chua L.O.The role of synchronization in digital communications using chaos—Part II: Chaotic modulation and chaotic synchro nization [J].IEEE Transactions on Circuits and Systems Part I, 1998, 45(4): 1129-1140.

[44] Lau F.C.M.,Tse C.K.Optimum correlator-type receiver design for CSK communication systems[J].International Journal of Bifurcation and Chaos, 1998,12(5):1029-1038.

[45] Ljupco Kocarev,Goce Jakimoski.Logistic map as a block encryption algorithm [J].Physics Letters A, 2001,289(4-5):199-206.

[46] Goce Jakimoski,Ljupco Kocarev.Chaos and cryptography:Block encryption ciphers based on chaotic maps[J].IEEE Trans. Circuits and Systems-I,2001, 48(2): 163-169.

[47] Goce Jakimoski,Toni Stojanovski,Ulrich Parlitz.From chaotic maps to encryption schemes[C].Proc.IEEE Int. Symposium Circuits and Systems 98, IEEE, 1998(4):514-517.

[48] Jesús Uís, Edgardo Ugalde,Gelasio Salazar.A cryptosystem based on cellular automata[J].Chaos,1998,8(4):819-822.

[49] Shujun Li, Xuan Zheng, Xuanqin Mou, and Yuanlong Cai. Chaotic encryption scheme for real-time digital video[C].Real-Time Imaging VI, Proceedings of

SPIE,2002(4666),149-160.

[50] Bruce Schneier.Applied Cryptography--Protocols, algorithms, and souce code in C[M]. John Wiley & Sons, Inc.,New York, second edition, 1996.

[51] B.V.Chirkikov,F.Vivaldi.An algorithmic view of pseudochaos[J].Physica D,1999,129(3-4):223-235.

[52] Marco Götz, Kristina Kelber, Wolfgang Schwarz.Discrete-time chaotic encryption systems--Part I: Statistical design approach [J].IEEE Trans.Circuits and Systems--I, 1997, 44(10):963-970.

[53] Christopher F., Woodcock, Nigel P. Smart.P-adic chaos and random number generation [J]. Experimental Mathematics, 1998, 7(4):333-342.

[54] Masaki Miyamoto, Kiyoshi Tanaka, Tatsuo Sugimura.Truncated baker transformation and its extension to image encryption[C].Mathematics of Data/Image Coding, Compression, and Encryption II, Proceedings of SPIE.1999 (3814):13-25.

[55] Mieczysław Jessa. Data encryption algorithms using one-dimensional chaotic maps[C].Proc. IEEE Int. Symposium Circuits and Systems 2000, IEEE,2000(I): 711-714.

[56] Th. Beth, D. E. Lazic, A. Mathias.Cryptanalysis of cryptosystems based on remote chaos replication[C].In Advances in Cryptology-EuroCrypt'94, Lecture Notes in Computer Science .Spinger-Verlag,Berlin,1994(0950):318-331.

[57] Kevin M. Short.Signal extraction from chaotic communications [J]. Int. J.Bifurcation and Chaos, 1997, 7(7):1579-1597.

[58] Chang-Song Zhou, Tian-Lun Chen.Extracting information masked by chaos and contaminated with noise: Some considerations on the security of com -munication approaches using chaos [J].Physics Letters A,1997,234(6):429-435.

[59] Tao Yang, Lin-Bao Yang, Chun-Mei Yang.Cryptanalyzing chaotic secure communications using return maps[J].Physics Letters A,1998,245(6):495-510.

[60] Maciej J. Ogorzatek, Hervé Dedieu.Some tools for attacking secure Communication systems employing chaotic carriers[C].Proc.IEEE Int. Symposium Circuits and Systems 98, IEEE, 1998(4):522-525.

[61] J. C. Yen,J. I. Guo.A new chaotic mirror-like image encryption algorithm and its vlsi architecture[J].Pattern recognition and image analysis,2000,10(2): 236- 247.

[62] 易开祥,孙鑫,石教英.一种基于混沌序列的图像加密算法[J].计算机辅助设计与图形学学报, 2000,12(09): 672-676.

[63] E.Alvarez, A.Fernandez, P. Garci.New approach to chaotic encryption[J].Phys Lett A,1999(263):373-375.

[64] Robert A.J.Matthews.On the derivation of a "chaotic" encryption algorithm [J].Cryptologia,1989,XIII(1):29-42.

[65] 周红,凌燮亭.有限精度混沌系统的 m 序列扰动实现[J].电子学报,1997,25(7): 95-97.

[66] A. Palacios ,H.Juarez.Cryptography with cycling chaos[J]. Physics Letters A, 2002,303(5-6):345-351.

[67] Kwok-Wo Wong.A fast chaotic cryptographic scheme with dynamic look-up table[J].Physics Letters A, 2002,298(4):238-242.

[68] P.García, A.Parravano, M.G. Cosenza, J.Jiménez, A. Marcano.Coupled map networks as communication schemes[J].Physical Review E, 2002,65(4):45201.

[69] P.García,J.Jiménez.Communication through chaotic map systems[J].Physics Letters A, 2002,298(1):34-40.

[70] Patrik Ekdahl,Thomas Johansson.Another Attack on A5/1[J].IEEE Transactions on Information Theory.2003:49(1):284-289.

[71] 潘松,黄继业,王国栋.现代 DSP 技术[M].西安:西安电子科技大学出版社.2003:57-91.

[72] Shujun Li. Analyses and New Designs of Digital Chaotic Ciphers [D].Ph.D. Dissertation, Xi'an Jiaotong University, 2003.

[73] 周红.一类混沌密码序列的设计方法及其在有限精度实现问题分析[D].博士论文.复旦大学,1996.

[74] Naoki Masuda,Kazuyuki Aihara.Dynamical characterstics of discretized chaotic permutations [J].Int. J. Bifurcation and Chaos, 2002,12(10):2087-2103.

[75] B.V.Chirkikov,F.Vivaldi.An algorithmic view of pseudochaos[J].Physica D, 1999,129 (3-4):223-235.

[76] Y.E.Levy.Some remarks about computer studies of dynamical systems[J]. Physics Letters A, 1982, 88(1):1-3.

[77] Slawomir T.Fryska,Mohamed A.Zohdy.Computer dynamics and shadowing of chaotic orbits [J].Physics Letters A, 1992, 166(5-6):340-346.

[78] Francois Robert.Discrete Iterations:A Metric Study[M].Springer-Verlag, Berlin:Springer Series in Computational Mathematics,1986.

[79] 海康威视.DS-4000HC、HCS、HC+、HF、HS、MD 系列板卡 SDK 说明书, Version 4.3[M].杭州海康威视有限公司,2008.6.

致 谢

三年的研究生时光转瞬即逝，在此论文完成之际，首先要向三年来始终关怀我、指导我、帮助我的丁群老师表达我诚挚的感谢和由衷的敬意！在三年的求学路途上，丁老师无论是学业上还是生活上都给予我莫大的帮助。在学习上，丁老师以其严谨的治学精神感染并教育着我，使我无论是在理论上还是实际应用中都获益匪浅，从毕业论文的选题到项目实践直至论文的写作，丁老师始终仔细指导，帮助我不断完善。在生活中，丁老师更是给予我很大的关怀和照顾，对我犹如母亲一样的呵护，令我感激不尽。丁老师对工作的投入，废寝忘食的科研精神以及善良平和的待人之道将会是我一生学习的榜样！在此，祝愿恩师工作顺利，身体健康，阖家幸福！

在此特别感谢香港城市大学电子工程系的陈关荣教授以及 K.W.Wong 教授，感谢他们对我论文研究工作的大力支持和帮助，感谢黑龙江大学电子工程学院的朱勇老师、张忠老师、杨自恒老师以及所有关心我、教育我的老师们，你们多年来对我的教育与培养，你们的一言一行都将令我受益终身，我的成长离不开你们的教诲！

还要感谢三年来的同窗好友周平、叶爱平，617 实验室的所有师兄、师姐、师弟、师妹们以及寝室的兄弟们，感谢你们对我学习和生活的大力帮助，这三年的学习生活能够和你们成为好朋友是我一生的宝贵财富，祝愿大家都学业有成、生活美满幸福！我们永远是一家人！

最后将最深的感谢和祝福献给我亲爱的父母，从你们身上我看到了天下最无私的爱，20 余载的含辛茹苦，你们饱经风霜！感谢你们对我的养育和栽培之恩，正是因为你们无私的呵护我、抚养我、培养我，我才能够走完近 20 年的求学之路，顺利完成学业。谢谢你们！祝愿你们身体健康，幸福快乐，儿子永远爱你们！

谨以此文献给他们！

攻读硕士学位期间参加的科研项目与成果

1、国家自然科学基金项目.基于 FPGA 技术的混沌序列密码加密芯片的研究.（No.60672011）.2007.1~2009.12.参与者.

2、香港城市大学基金项目.A Fast Encryption and Authentication Scheme based on Chaotic Maps. (Project no.:7002070).Certificate of Service：Research Associate employed by City University of Hong Kong for the period of 2008.9~2008.12.

3、2009 年“挑战杯”黑龙江省大学生课外学术科技作品竞赛三等奖.基于 SOC 的网络加密卡.2009.11.

4、黑龙江大学学生学术科技创新项目.视频采集加密系统的研究与构建.2009.3~2009.12.项目负责人。