一种混沌流密码算法设计与实现

董斌辉,周健勇
DONG Bin-hui,ZHOU Jian-yong

上海理工大学 管理学院,上海 200093
College of Management,University of Shanghai for Science and Technology,Shanghai 200093,China
E-mail:asxinyu@126.com

DONG Bin-hui,ZHOU Jian-yong.Design and implementation of chaos based stream cipher.Computer Engineering and Applications,2009,45(35):120-122.

Abstract: This paper proposes a new kind of stream cipher encryption algorithm that is based on the Logistic chaos mapping. And this algorithm describes a new method of chaos disturbance using the random properties of chaos system.Through encoding algorithm and on the basis of the numeralization of chaos random sequence,a new non-linear transformation algorithm is introduced in order to resist various attacks of the chaotic stream cipher system.The statistical tests and related analysis prove that the key series has high complexity and good cipher quality.The entire encryption system has long period and good flexibility,the encryption model can be extended to other more complex chaos systems.

Key words: chaos encryption;stream cipher;non-linear transformation;statistical tests

摘 要:提出了一种基于 Logistic 混沌映射的流密码算法,该算法利用混沌本身所具有的随机特性,提出了一种新的对混沌系统扰动的方法。通过编码算法以及在混沌随机序列数字化的基础上引入一种新的非线性变换算法,以抵抗对混沌流密码系统的各种攻击。经统计测试和相关分析,密钥序列具有较高的线性复杂度和良好的密码学特性。整个加密系统的周期性大、灵活性好,加密模型还可以推广到其他混沌系统。

关键词:混沌加密;流密码;非线性变换;统计测试
DOI:10.3778/j.issn.1002-8331.2009.35.036 **文章编号**:1002-8331(2009)35-0120-03 **文献标识码**:A **中图分类号**:TP309.7

1 引言

混沌是一种貌似无规则、在确定性系统中出现的一种对初值非常敏感的类似随机过程,它是非线性动力学系统具有内置随机性的一种表现[1]。因为混沌系统迭代产生的时间序列对初始条件敏感、结构复杂难以分析和预测,而且可以提供具有良好随机性、复杂性的长周期伪随机序列,这些特性都是流密码的基本设计要求,因此完全可以将混沌应用于流密码系统。混沌加密的基本原理就是利用混沌系统产生的混沌序列经过变换后得到的密钥流对明文加密,接收方利用同样的混沌系统和相同的参数产生同样密钥流对密文解密;加密用的密钥序列并不需要传递,只需要将混沌系统类型及相关初始参数作为密钥进行传递就可以了;而混沌密码系统的加密端和解密端是两个独立的完全相同的混沌系统。提出的混沌流加密算法采用了基于混沌序列本身的扰动算法来克服其有限精度效应带来的问题,同时采用编码算法和非线性变换来进一步提高混沌系统和输出序列的随机性,克服了平凡混沌加密带来的极不安全性,该算法产生的随机序列周期理论上至少可以达到 10^{600} 以上,完全满足实际应用要求。

2 混沌加密算法设计

属于流密码范畴的混沌加密系统,其关键在于随机数的产生[2-4],同时要考虑到一些针对混沌流密码的攻击手段[5-7]。提出的基于混沌的加/解密系统所用的随机数是将两个低维混沌动力系统进行迭代计算,利用动态编码表作为非线性变换来生成伪随机数序列,然后利用伪随机数序列作为密钥对明文进行加密。其基本原理如图 1(解密过程与加密过程完全相同)。

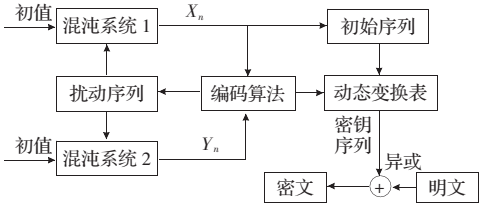

图 1 混沌系统加密基本模型

2.1 克服数字混沌有限精度效应的方法

理想的混沌系统其状态是无限不重复的,但在实际应用中由于计算机字长和精度的限制,迭代序列会偏离实际的混沌轨

作者简介:董斌辉(1984-),男,研究生,主要研究领域:系统分析与优化,信息安全;周健勇(1970-),男,副教授,主要研究领域:系统优化、系统工程、信息安全。

收稿日期:2008-09-19 **修回日期**:2008-12-25

道。对于精度为 L 的数字混沌序列,其周期一定不大于 2^L,故在应用时必须解决此类有限精度效应问题。文献[8]总结了很多提高混沌周期的改进方法,考虑下列 Logistic 系统扰动方程:

$$x_{n+1}=\mu_n \cdot x_n(1-x_n)+\Delta_n$$

提出的改进算法是基于对 μ_n 和 Δ_n 的扰动来生成混沌序列,该方法充分利用了混沌序列的随机特性,利用与其他混沌系统产生的序列结合,采用相关算法提取随机参数作为混沌系统的扰动序列,对混沌序列施加扰动的原理如图 2 所示。

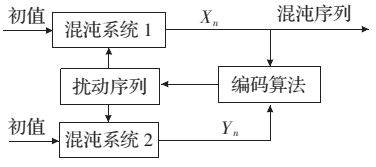

图 2 混沌系统扰动原理

如图 2 所示,混沌系统 1 为主混沌系统,用来产生初始密钥序列,而混沌系统 2 作为辅助系统产生 Y_n 序列,通过编码算法提取扰动序列,为提高整个系统的周期性,应对两个混沌系统都施加扰动。其中扰动序列是通过当前的混沌序列 X_n 与 Y_n,利用编码算法来产生的。编码算法的相关参数可以作为密钥,即使获得了系统的初始值,若不能得到扰动的相关参数,也同样不能得到正确的密钥序列,因为混沌系统对初值的高度敏感性,加上扰动等多种随机因素的影响,轨道不断地在进行变换,系统很快就会进入新的混沌轨道。

2.2 编码算法及混沌序列数字化

编码算法不仅涉及对两个混沌系统的扰动,而且与后面的动态变换表的变化有很大关系。该文介绍的编码算法目的是提取扰动序列,以及提取动态变换表所需的参数,其实质是对混沌序列进行采样、变换然后产生相关序列及参数。算法的部分细节可以灵活扩展、修改,在此并没有详细规定。

步骤 1 初始化混沌系统 1、2 的参数,并舍弃前若干次迭代的混沌序列;设当前的混沌序列为 x_n 和 y_n。

步骤 2 从当前的混沌系统输出 x_n,y_n 中分别提取 $m/2$ 位数字(m 为偶数),提取规则作为密钥给出。设提取的数字分别为 $A_n=\{a_1,a_2,\cdots,a_{m/2}\}$,$B_n=\{b_1,b_2,\cdots,b_{m/2}\}$;将两个序列按一定规则组合成序列 $C_n=\{c_1,c_2,\cdots,c_m\}$;若 c_1 为 0,则舍弃当前的混沌序列,继续迭代混沌系统或调整组合规则。

步骤 3 由 C_n 根据下列公式求出下一次混沌迭代的扰动变量 μ_n 和 Δ_n,并将 μ_n 变换到混沌区域内:

$$\begin{cases} V_n=\sum_{i=1}^{m} c_i\times 10^{2-i},\ \mu_n=V_n-\left\lfloor \frac{V_n-\mu_2}{\mu_2-\mu_1}\right\rfloor\times(\mu_2-\mu_1),\ \mu_1=3.569\,945\,68,\ \mu_2=4 \\ \Delta_n=\alpha\times x_n+(1-\alpha)\times y_n\ (\alpha \text{ 为系统参数,由密钥给出}) \end{cases}$$

步骤 4 继续迭代混沌系统 1,2,将步骤 3 中求得的 μ_n 和 Δ_n 做为扰动参数,将输出结果进行下列变换,使其在指定范围内,即得到下一次的混沌序列:

令 $temp=\mu_n \cdot x_n(1-x_n)+\Delta_n$

$$x_{n+1}=\begin{cases} temp, & temp<1 \\ temp-\lfloor temp\rfloor, & temp\geqslant 1 \end{cases}$$

步骤 5 Y_n 的扰动与 X_n 扰动序列的生成算法一样,不同在于参数的选择。

上述编码算法涉及对混沌序列的数字化处理,由于采用经迭代直接生成的实数序列的平凡混沌加密容易破解,而且考虑到二值量化会损失混沌序列的伪随机性,该文采用的数字化处理是随机选取混沌序列中的有效数字组合,构成整数并对 256 取余,得到 8 位二进制序列。不仅增强了其随机性,而且还增大了密钥空间。

2.3 动态变换表

动态变换表是为了增强系统的非线性,提高输出序列的随机性和复杂性以抵抗针对混沌加密的各种攻击所采取的一种算法。文献[9]指出,对一般的混沌编码模型,通过对其的符号动力学分析,就能以较高的精度很快估计其根密钥,因此必须在混沌编码模型产生的符号序列作为伪随机性序列输出之前引入非线性变换,这是设计高安全性数字混沌密码系统的关键技术之一。该文采用的动态变换表算法不仅极大地提高了系统的周期性,并有效地改善了序列的随机性能,同时该算法还可以作为一种全排列生成算法。实验表明:将混沌序列数字化处理后得到的整数作为变换的输入,输出序列的随机性得到显著的增强。算法原理如下:

步骤 1 初始化:记一维数组 table 大小为 256,初始化为 $table[i]=i$,$0\leqslant i\leqslant 255$;其输入值为一字节的正整数(0~255);

步骤 1.1 令从 $i=0$ 开始,然后根据编码算法从混沌序列 X_n,Y_n,每次提取一个 0~255 内均匀分布的整数 k(若 $i=k$,则重新取 k 值)。将 $table[i]$与 $table[k]$的值互换,直到 $i=255$,一轮变换完成;将此过程称为一轮完全变换。由于变换表初始化很重要,关系到输出序列的随机性,可以根据实际情况,调整初始化轮数。建议初始化轮数不少于 20~40 轮。

步骤 1.2 从上述一轮完全变换可以看出,表中每个元素平均要变换两次位置。为增强变换随机性并考虑算法效率,引入一种新的部分随机变换,规则如下:根据编码算法从混沌序列 X_n,Y_n,每次提取 2 个 0~255 内的整数 $k1$ 和 $k2$(若 $k1=k2$,则重新提取 $k1$,$k2$,直到其不相等);将 $table[k1]$与 $table[k2]$的值交换,此过程称为一次基本变换,若将变换表同时施加 n 次基本变换,称为一次部分变换。建议 n 不小于 20 次。

步骤 2 变换表的输入输出:若输入值为 $value$,则输出为 $table[value]$;

步骤 3 在输出值(作为加密的密钥流)之后,应对变换表进行动态变换,为下一次提取密钥做准备。具体的变化可以根据编码算法产生参数,或者设定每次的变换规则。如可以每次都进行一次部分变换,或者在提取若干字节密钥后,进行若干轮完全变换,该过程的相关参数可以作为密钥,即使取得了混沌的初始参数,变换表的参数错误,也是不能够正常解密的。

3 算法分析与检验

混沌流密码加密系统的安全性关键在于其密钥流序列的随机性,不仅要求其随机序列满足均匀独立的分布,而且也有一定的周期性要求。提出的混沌加密系统具有下列优点。

(1)系统可灵活扩展、改进。由图 1 所示的原理及编码算法可知,影响加密系统的因素很多,而且任意一个参数的极小变化都将迅速扩散到整个混沌系统,影响到输出序列。各模块之间均由相应的密钥控制其变化,相互独立;但各模块对序列的处理均影响到最终密钥序列,因此又相互紧密联系。所以,可以根据实际情况自由选择密钥长度、以及对各模块进行灵活扩展与改进,也使得攻击的难度大大增加。

(2)混沌系统的非周期性。实际的混沌系统具有非周期性,

但是数字混沌由于计算机的限制而具有短周期性,极其不安全。而采用的参数扰动的方法极大改善了数字混沌系统的周期性。考虑一般编程语言的双精度型数据类型进度可以到达 14~15 位小数,甚至更高。当取各项初始参数为 14 位小数位时,则初步估算混沌系统 1 的周期至少为 10^{100}~10^{110}（不仅依赖混沌系统 2,而且还与编码算法有关,为各因素周期的最小公倍数）;28 位小数时,周期为至少为 10^{200} 以上。

(3)密钥序列的非周期性。引入的非线性变换——动态变换表的周期性不仅依赖于输入序列(混沌序列)的周期性(L)相关,而且与自身变换的周期性相关,即其周期性理论上达到:$L\times 256!$,即使不考虑 L 的影响,其周期也可以到达 10^{506} 以上。理论上,周期性的输入序列通过动态变换表后将不能得到周期性的输出序列,因为动态变换表的周期近乎无穷大。其周期性完全满足实际应用的要求。

(4)随机性检验。为考察混沌序列(变换前)及密钥序列(变换后)的随机性,对两种序列进行了一系列统计测试。对两种序列采用 5 种基本统计测试。下面是测试结果(注:测试长度为 20 000 位,测试次数均为 10 000,每组数据都单独产生。表中所记录的为通过该项检验的次数,其中总计项为通过所有测试项目的组数):

表 1　随机性测试结果

	基本统计测试(通过次数)(显著水平 0.05)					
	频率测试	序列测试	扑克测试	游程测试	自相关测试	总计
混沌序列	5 241	4 727	5 217	7 665	9 747	2 560
变换后序列	9 445	9 442	9 403	9 306	9 750	8 067

(5)线性复杂度分析。采用 Berlekamp-Massey(BM)[10]算法计算了变换后随机序列的线性复杂度,结果表明密钥序列具有良好的线性复杂度轮廓。下面是变换后序列的线性复杂度轮廓曲线。

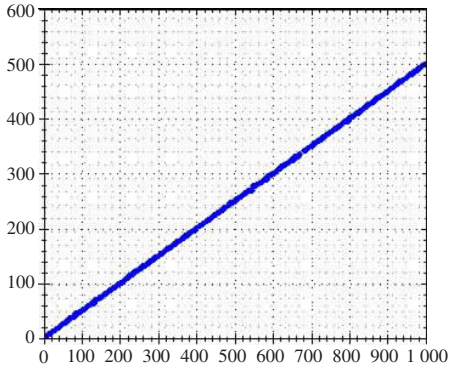

图 3　密钥序列线性复杂度轮廓

从测试数据可以明显看到,经过动态变换表后的随机序列的随机性明显要好于变换前的混沌序列。可以看出,变换后的随机序列具有很好的统计特性,从一定程度上反应了算法的安全性。而且变换后序列也具有良好的线性复杂度特性。

因此从上述分析以及流密码的特点可知,该加密系统具有抗穷举攻击分析、抗统计攻击分析的能力;通过加密实例测试结果显示,由于加密系统对密钥等参数的高度敏感性和采用了非线性变换表,使得加密系统还具有抗差攻击分析以及抗相空间重构攻击的能力。

4　算法实现与加密实例

采用 C# 语言在开发工具 SharpDevelop3.0 环境下实现了上述提出的混沌加密算法。正确的加、解密效果如图 5 所示。

图 5　正确加解密效果

如图 5 所示,明文为 4 个重复在短句“Hello,everyone!”,但其密文并不重复,而且能够正常解密。由于该加密系统利用混沌系统的特性,以及加密算法所采用的扰动和动态变化表策略,使得系统对密钥具有高度的敏感性,如图 6 所示,将解密密钥 $X0$ 改变 10^{-14} 后解密就出现了乱码,不能正常解密,由于加密算法中扰动和编码算法部分的多个参数都可以作为密钥,每一步都要改变其轨道状态,很快就会进入新的混沌轨道,再加上混沌数字化采取的方法特别,即使轨道相近,也不能得到正确的密钥序列。

图 6　密钥相差 e-14 时的解密效果

5　总结

混沌加密在其安全性方面有着独特的优势,其加解密速度快,安全性高。提出的加密模型适合用软件实现,能够用于文本文件、语音、图像等各种数据的加密,加密模型可以推广到其他高维或多维混沌系统,也可以采用不同的混沌系统进行组合,使得其安全性进一步提高。当然,该模型的许多细节还可以进一步加以研究改善,例如,两个混沌系统可以采用不同的混沌映射,文中的混沌系统都是采用 Logistic 映射,只是作为其最简单的一个实例。值得注意的是,混沌加密系统是不能够直接用初始参数等实数作为加密者的密钥来记忆的,因此,还需要研究能够将字母符号序列映射成混沌系统加密参数的方法,这样才能有实用性。

参考文献:

[1] 许国志.系统科学[M].上海:上海科技教育出版社,2000.

[2] Mao Y.A chip performing chaotic stream encryption[J].Studies in Computational Intelligence(SCI),2007,42:307-332.

[3] Li P.Securing communication by chaos-based encryption[J].Studies in Computational Intelligence(SCI),2007,42:285-306.

(下转 150 页)