一种基于符号动力学的伪随机序列发生器设计方法

张 伟[1,2] 韦鹏程[1] 杨华千[1,2]

(重庆教育学院计算机与现代教育技术系 重庆400067)[1] (重庆大学计算机科学与工程学院 重庆400044)[2]

摘 要 混沌序列具有带宽大、类噪声、难于预测和重构等特点,因而常使用于网络通信和数据加密。本文提出一种基于符号动力学和混沌系统的伪随机序列发生器的设计方法。为了克服计算机有限精度造成的短周期,对生成的混沌序列进行了周期性修正,这样不仅增大了序列的周期,而且使得序列的周期可度量。我们用两个混沌系统产生的符号序列进行异或运算后作为最终输出,从而进一步增加了输出序列的随机性,也使得输出序列更加难以破译。理论研究和模拟结果表明,该序列发生器具有较好的随机性并且便于软硬件的实现。

关键词 混沌系统,符号动力学,伪随机序列发生器

A Design Method of Pseudo-Random Sequences Generator Based on Symbolic Dynamics

ZHANG Wei[1,2] WEI Peng-Cheng[1] Yang Hua-Qian[1,2]

(Department of Computer and Modern Education Technology, Chongqing Education College, Chongqing400067)[1]

(Department of Computer Science and Engineering, Chongqing University, Chongqing400044)[2]

Abstract As we all know, chaotic sequence possess good features such as broad band, noise-like, accurate regeneration and difficult prediction, so it is suitable for information encryption. In this paper, Pseudo-Random Sequences Generator based on Symbolic Dynamics and chaotic system is presented. In order to overcome short period resulted from the finite precision in practical application, we amend the output of chaotic sequence periodically so as to lengthen the period of the sequence and make the period of the sequence can be measured. Moreover, the bit sequence generated by the XOR operation of two chaotic sequence is used as the final output sequence, by this method, the randomness of the output sequence will be increased thus difficult to decrypt. The computer simulation results also show that this Pseudo-Random Sequences Generator has good randomness properties and can be implemented easily in both software and hardware.

Keywords Chaotic system, Symbolic dynamics, Pseudo-RandomsSequences generator

1 引言

密码算法的安全性基于密钥的安全性,而不是基于算法的细节的安全性,而好的密钥是指那些自动处理设备产生的随机的位串。如果密钥为64位长,每一个可能的64位密钥必须具备相等的可能性,这些密钥位要么从可靠的随机源中产生,要么从安全的伪随机位发生器中产生。能否产生真正的随机数,长期以来这个问题都处在激烈的争论之中。但产生具有随机数统计特征并且不可再现的序列肯定是有益的。传统密码学中的几种方法如使用 RAND 表、随机噪声、计算机时钟等都是很好的方法,但这些方法不容易实现和控制[1]。

由于混沌系统对初始条件和系统参数非常敏感以及生成的混沌序列具有非周期性和类随机性的统计特性,近年来混沌系统在保密通信领域的应用得到了较多的研究,主要集中在混沌密码系统的设计[2~6]以及通信系统中混沌的同步和调制[4~8]等。我们通过差分方程的迭代运算来产生混沌序列,并采用符号动力学的思想从中产生符号序列。为了增加符号序列的随机性,我们对序列进行了随机扰动;为了消除计算机有限精度对安全性的影响,我们对序列进行了周期性修正,大大增加了实际输出序列的周期。

2 符号动力学和混沌系统

对于一个混沌系统,若不关心轨道点 x_i 的具体数值,而只根据 x_i 在相空间中的位置,把它与某个符号对应,即令每一个 x_i 对应一个符号 s_i,这样一条数值轨道就对应一个符号序列。混沌系统的动力学行为,例如遍历特性、对初值敏感特性、伪随机不可预测等,都可以由这个简单的符号序列完全刻画。一般来说,数值轨道与符号序列是多对一的。许多不同的数值轨道,可能对应同一个符号序列,而不同的符号序列,一定对应不同的数值轨道。我们正是利用混沌系统符号动力学的这一优良特性,设计一种伪随机序列发生器。

对于一维单峰映射(见图1):

$$x_{n+1}=f_p(x_n) \tag{1}$$

其中 p 是系统参数。符号动力学不关心系统轨道点的具体数值,而根据 x_k 的位置,把它与某个符号对应,即令每个 x_k 对应一个符号 s_k:

$$s_k=\begin{cases}R & x_k>C\\ C & x_k=C\\ L & x_k<C\end{cases}$$

这样对给定的初始点 x_0 由映射式(1)产生的数值轨道$\{x_k\}$:$x_0,x_1,\cdots,x_n,\cdots$就对应唯一的符号序列$\{s_k\}$:$s_0,s_1,\cdots,s_n,\cdots$。显然,数值轨道与符号序列可能是多对一的关系,但是符号动力学发现当系统的参数达到某一个极限值 p_∞ 时会出现 RL^∞ 型超稳揉序列,而任何不以 RL^∞ 结尾的符号序列均为允许字并与区间上的点一一对应。这正是我们设计伪随机序列发生

器的理论基础。

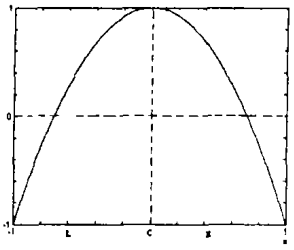

图1 单峰映射

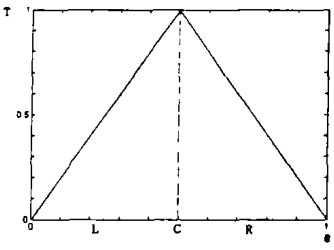

图2 帐篷映射

考虑 Logistic 映射(见图1)

$$x_{n+1}=f(x_n)=1-2x_n^2,\quad -1\leqslant x_k\leqslant 1 \tag{2}$$

和帐篷映射(见图2)

$$\theta_{n+1}=T(\theta_n)=\begin{cases}2\theta_n & 0\leqslant\theta_n<0.5\\ 2(1-\theta_n) & 0.5\leqslant\theta_n\leqslant 1\end{cases} \tag{3}$$

容易看出这两个映射都是满映射,并且已经证明 RL^∞ 为它们的超稳揉序列。从而对式(2)若令 $I=(-1,1)$,$S=\{$不以 RL^∞ 结尾的符号序列$\}$,则存在 I 到 S 的一一对应关系,映射式(3)也有类似结果。

可以证明映射式(2)和式(3)的迭代序列$\{\theta_k\}$和$\{x_k\}$具有以下统计特性:

(i)映射式(2)和式(3)拓扑共轭,从而由$\{\theta_k\}$的分布密度 $\rho_T(\theta)$可以推出$\{x_k\}$的分布密度 $\rho_f(x)$。事实上,由

$$\rho_f(x)=\rho_T(h^{-1}(x))\left|\frac{\mathrm{d}h^{-1}(x)}{\mathrm{d}x}\right|$$

易得

$$\rho_f(x)=\frac{1}{\pi\sqrt{1-x^2}},\quad -1<x<1$$

其中 $h(\theta)=-\cos(\pi\theta)$,$0\leqslant\theta\leqslant 1$,为 f 到 T 的拓扑共轭映射。

(ii)它们的均值及 m 阶互相关函数分别为

$$\bar{x}=\int_{-1}^{1}x\rho_f(x)\mathrm{d}x=0\quad \bar{\theta}=\int_0^1\theta\rho_t(\theta)\mathrm{d}\theta=\frac{1}{2}$$

$$C_f(m)=\int_{-1}^{1}\int_{-1}^{1}\rho_f(x,y)(x-\bar{x})(f^m(y)-\bar{y})\mathrm{d}x\mathrm{d}y=0$$

$$C_T(m)=\int_0^1\int_0^1\rho_T(x,y)(x-\bar{x})(T^m(y)-\bar{y})\mathrm{d}x\mathrm{d}y=0$$

而自相关函数类似于 δ 函数。

由此可见,混沌序列$\{\theta_k\}$和$\{x_k\}$具有良好的随机性。由前面对符号动力学的讨论知,与混沌序列对应的符号序列也具有良好的类随机性。我们改变符号 s_k 的定义:

$$s_k=\begin{cases}1 & x_k>C\\ C & x_k=C\\ 0 & x_k<C\end{cases}$$

从而,符号序列就成为 0-1 序列(注:符号为 C 时要进行修正,见本文第 3 节)。

3 伪随机序列发生器的设计

从理论上讲,对任意给定的初始值 x_0和 θ_0($x_0\neq 1/2,1$;$\theta_0\neq 0,2/3$)迭代方程(2)和(3)会分别产生一个非周期的无穷数值序列,从而对应一个无穷随机序列。由第2节的讨论知,该随机序列也是非周期的和类随机的。但计算机精度有限使得实际产生的序列必然具有周期性。事实上,我们无法消除周期性[11~12],而只能设法延长序列的周期。为此我们提出一种新的思想,即周期性地对迭代系统的状态进行修正。这使得输出序列的周期大大增加,并可以加以度量。例如,修正周期为 T_1,修正系统的周期为 T_2,则输出序列的周期为 T_1*T_2。如果你要得到更长的周期,还可以对修正系统进行修正。在实际设计时,我们考虑到以下控制技术:

(i)去掉每次迭代的前有限项及每次修正后的前有限项(比如前50项);

(ii)采用时间延迟技术;

(iii)若初始值选取不当使得产生的序列中出现字符 C 时,及时对系统状态进行修正;

例如对迭代方程(3)若取 $x_0=0.5$,则产生序列 $C100\cdots$;

(iv)从图3可以看到,我们把内部状态的输出位又与另一序列发生器的输出位进行了异或运算,这样做的目的在于增加输出序列的随机性。序列发生器 G2可以有与 G1完全相同的结构,不过在这里我们采用线性映射-帐篷映射产生随机序列。

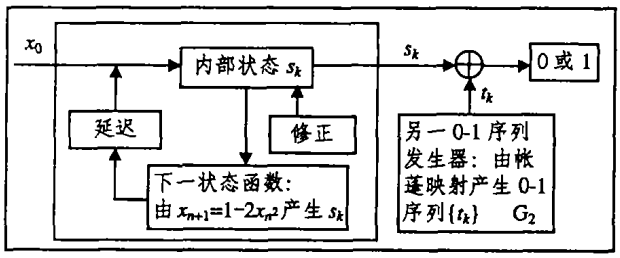

图3 0-1序列发生器模型

4 性能分析及对输出结果的数值模拟

首先我们分析系统的抗干扰性能。从混沌序列中任意截取一段序列记为 $S=\{s_{k_1},\cdots,s_{k_m}\}$,设在传输中噪声 $E=\{e_1,\cdots,e_m\}$加入到输出序列中,则实际输出序列为:$R=S+E$。如果我们把混沌序列 S 和实际输出序列 R 看成 m 维向量,那么相关系数 C 等于 S 和 R 的数量积,即

$$C=S\cdot R=S\cdot S+S\cdot E$$

由于混沌序列的相关函数具有快速衰减特性,因此这些噪声与现有的序列是不相关的,即 $S\cdot E$ 很小。系统对于附加噪声和乘性干扰传输是鲁棒的。

下面我们再分析系统的抗破译能力。由于系统具有足够长的周期,所以抗生日攻击和穷举攻击的能力强[10];由于混沌系统的类随机特性,已知明文攻击和选择明文攻击是不可能的;对一些弱混沌系统甚至有多个正 Lyapunov 指数的超混沌系统,从理论上讲,破译者都可以通过神经网络重构出来,只是算法所能达到的速度和精度问题。但本文采用了两个不同的混沌系统,它们相互作用后才能产生输出。而且,混沌序列的产生中已经加入了舍入误差,混沌系统已不完全遵守系统方程,因此这就增加了系统重构的难度。同时,混沌序列产生无穷多种序列,而编码又有很多种方法,因此抗破译能力较强。我们的数值模拟结果如下:

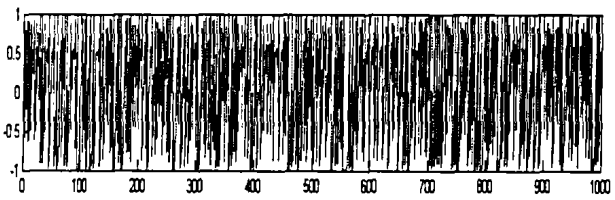

图4 Logistic 映射产生的混沌序列分布函数

(下转第180页)

非常昂贵的过程。也就是说整个过程中，系统大量时间耗费在二进制编码的大整数到十进制的转换上，最终导致 BigDecimal 的运行时间也达到了1953ms。最后我们将 BigDecimal 和 DecFlt 分别应用到实际的 Cobol 事务处理程序中，发现针对5000组不同测试样本 DecFlt 仍然较 BigDecimal 格式快20%左右。

表3 多精度编码算术比较

编码方案	DecFix	BigDecimal	DecFlt
运行时间(ms)	3282	1953	984
Hot 操作	数组乘，47.82%	幂次，32.05%	数组乘，25%
时间比	3.33	1.98	1
运行时间(ms)	n/a	3172	2641
相关 Hot 操作	n/a	除法，2.94%	数组乘，1.41%
时间比	n/a	1.20	1

结论 十进制算术的强烈需求和大多数嵌入式处理器中浮点部件的缺乏使得采用软件方式支持定浮点运算成为必需。本文通过对小数定浮点特性的深入分析，指出了在各种应用要求限制下的选择策略，并且对多精度的定浮点运算给出了以十进制编码系数的数据模型及其详细的算法，实验表明这种数据模型比 Java 中以二进制编码系数的 BigDecimal 数据类型性能提高近一倍。同时我们将此数据模型和算法运用在 Cobol2Java 翻译系统中，将实际的事务处理程序性能提高20%左右。

参 考 文 献

1 Perkins R. EASIAC, A Pseudo-Computer. Communications of the ACM, ACM, 1956, 3(2): 65～72
2 Jones F B, Wymore A W. Floating Point Feature on the IBM Type 1620. IBM Technical Disclosure Bulletin, 05-62, IBM, May, 1962. 43～46
3 Chen T C, Ho I T. Storage-Efficient Representation of Decimal Data. ACM, 1975, 18(2): 49～52
4 Stevenson D, et al. IEEE 754-1985 IEEE Standard for Binary Floating-Point Arithmetic. IEEE, July 1985. 20
5 Cody W J, et al. IEEE 854-1987 IEEE Standard for Radix-Independent Floating-Point Arithmetic. IEEE, March 1987. 14
6 Cowlishaw M F, Schwarz F M, Smith R M, Webb C F. A Decimal Floating-Point Specification. In: Proc. of the 15th IEEE Symposium on Computer Arithmetic, ISBN 0-7695-1150-3, IEEE, June 2001. 147～154
7 Cowlishaw M F. Decimal Floating-Point: Algorism for Computers. In: Proc. of the 16th IEEE Symposium on Computer Arithmetic (ARITH'03), 2003
8 Melton J, et al. ISO/IEC 9075: 1992: Information Technology - Database Language - SQL. ISO, 1992. 626
9 Shibamiya A. Decimal arithmetic in applications and hardware. pers. Comm. June 2000. 2
10 Knuth. The Art of Computer Programming. Addison-Wesley Publishing Company, 1981, 2: 269～271

（上接第141页）

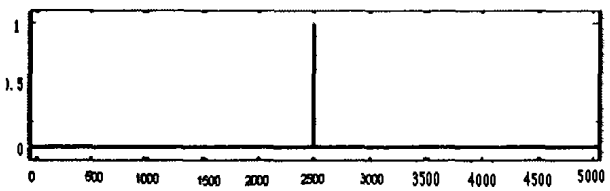

图5 输出序列的自相关函数

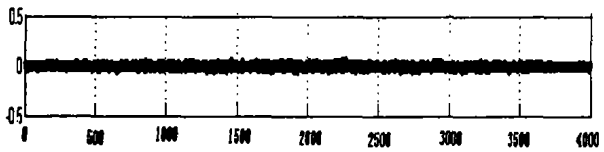

图6 输出序列的互相关函数

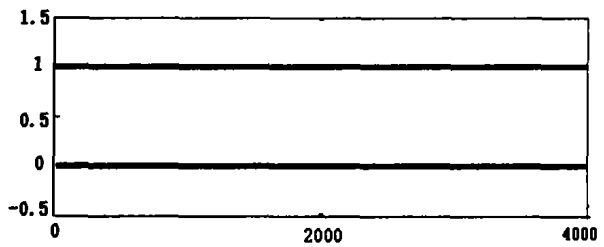

图7 输出序列的0、1分布

结论 本文通过 Logistic 映射和帐篷映射及相关控制技术提出了一种随机性较好的伪随机序列发生器的实现方法。当然只要对该方法稍加修改就可以得到任意位长的位串（密钥）发生器。在使用密码系统通信时，保持加密端和解密端的密钥序列发生器同步是至关重要的。在信息传递过程中，如果一个发生器跳过一个周期或者一个密文位在传输中丢失都可能出现解密错误，因此必须保证在继续进行之前使两个发生器重新同步。关于发生器的同步可参阅文[1,4,7,8]。

参 考 文 献

1 Rueppel R A. Linear complexity and random sequences. In: Advances in Cryptology. EURO CRYPT'85 (LNCS 219), 1986. 167～188
2 Sushchik M, Tsimring L S, Volkovskii A R. Performance analysis of correlation-based communication schemes utilizing chaos. IEEE Trans. on CAS-I FTAA, 2001, 47(12): 1684～1691
3 Gernak J. Digital generators of chaos. Phys. Lett. A, 1996, 214: 151～160
4 Frey D R. Chaotic digital encoding an approach to secure communication. IEEE Trans, On circuits and systems (II), 1993, 40(10): 660～666
5 Habutsu T, Nishio Y, Sasase I. et al. A secret cryptosystem by iterating a chaotic map. In: Advance in cryptology - EUROCRYPT'91, D. W. Davies, ed. LNCS 547, (Springer - Verlag, Berlin), 1991. 127～140
6 Kocarev L. Chaos-based cryptography: A brief overview. IEEE Trans. on CAS-I, 2001, 1(3): 6～21
7 Pecora M, Carroll L. Synchronization in chaotic systems. Phys. Rev. Lett., 1990, 64(8): 821～823
8 Cuomo K M, Oppenhein A V, Strogatz S H. Int. J. Bifurcation and Chaos[M]. Sci. Am., 1993, 269: 62～68
9 Robinson C. Dynamical systems: stability, symbolic dynamics and chaos[M] (2nd Edition). CRC, 1998
10 Jung O, Rul C. Encryption with Statistical Self-Synchronization in Synchronous Broadband Networks. In: Proc. Cryptographic Hardware and Embedded Systems, CHES '99: 1999. 340～352
11 饶妮妮. 一类混合混沌序列及其性能分析. 电子科技大学学报, 2001, 20(2): 115～119
12 周红. 有限精度混沌系统的 m 序列扰动实现. 电子学报, 1997, 25(7): 95～97