一种新的基于时空混沌的单向 Hash 函数构造*)

王 永[1,2] 廖晓峰[2] 杜茂康[1]
(重庆邮电大学电子商务与现代物流重庆市重点实验室 重庆 400065)[1]
(重庆大学计算机科学与工程学院 重庆 400044)[2]

摘 要 对时空混沌中常用的耦合映像格子模型(coupled map lattice,CML)和已有的基于时空混沌的单向 Hash 函数构造进行了分析,在此基础上提出了一种新的基于时空混沌的单向 Hash 函数构造方法。首先根据 Lyapunov 指数谱确定 CML 中参数的取值,然后用线性变换后的消息来更改 CML 的状态,并通过迭代来扩散消息中每个字节对 CML 状态的影响,Hash 值从最终的 CML 状态中抽取。研究结果表明,该方法具有很好的单向性、弱碰撞性、初值敏感性和灵活性以及更高的计算效率。

关键词 时空混沌,Hash 函数,耦合映像格子,Lyapunov 指数谱

Novel One-way Hash Function Construction Based on Spatiotemporal Chaos

WANG Yong[1,2] LIAO Xiao-feng[2] DU Mao-kang[1]
(Key Laboratory of Electronic Commerce and Modern Logistics of Chongqing Province, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[1]
(College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China)[2]

Abstract The CML(coupled map lattice) model and some algorithms for one-way Hash function construction based on spatiotemporal chaos was analyzed. A novel Hash function construction method based on spatiotemporal chaos was proposed. The parameters of CML are fixed based on the Lyapunov exponent sectrum. Then uses the linearly transformed message to change the state of CML and expands the influence of each byte in the message to the state of CML by iteration. The Hash value is extracted from the final state of CML. Simulation results show this method possesses the advantages such as good one-way, weak collision, sensitivity to initial values, flexibility and better calculating efficiency than the existed Hash function based on the spatiotemporal chaos.

Keywords Spatiotemporal chaos, Hash function, Coupled map lattice, Lyapunov exponent sectrum

1 引言

Hash 函数是密码学中一项重要基本技术,它在完整性验证、身份认证和数字签名等安全技术中都有广泛的应用[1]。近年来,出现了一些新的构造 Hash 函数的方法。基于混沌的 Hash 函数构造[2-8]是其中的热点之一。从已有的一些基于混沌的 Hash 函数的构造方案来看,存在如下不足:1)当 Hash 函数基于低维的混沌系统构造时,可利用混沌预测技术[9,10]对其进行分析,保密性能堪忧;2)在实际实现中,由于受到计算机有限精度的影响,混沌序列会退化为大周期序列,这对 Hash 函数的安全性有极大的影响。因此,提高混沌信号的复杂性和减少计算机有限精度影响是提高基于混沌的 Hash 函数安全性能的关键。

时空混沌系统属于高维的混沌系统,其内部任何一点的状态不仅与该点前一状态相关,而且还与它在系统中的位置有关。时空混沌系统中存在多个正的 Lyapunov 指数,能很好保证系统复杂的动态特性。另外,在时空混沌内部由于各点之间耦合在一起,相互影响,能极大地增加整个系统在有限精度下的周期。通常实际应用不会超出这样的大周期范围,因此可有效减少计算机有限精度的影响[11]。

为此,本文对已有的基于时空混沌的 Hash 函数进行了分析,在此基础上提出了一种新的 Hash 函数构造算法。理论分析和仿真试验表明,该算法除具有很好的统计特性外,还具有很高的效率,能有效减少产生 Hash 值的运算时间。

2 时空混沌构造 Hash 函数的可行性

单向函数的定义:映射 $H: X \to Y$ 对所有的 $x \in X$, $H(x)$ 容易计算,但其逆过程计算困难,即给定 $H(x)$ 计算 x 困难。这样的函数称为单向函数。单向 Hash 函数是一种特殊的单向函数,具有如下 4 条性质[12]:

1)输入为任意长度的序列,输出为固定长度的 0,1 序列;

2)不可逆性:已知 c = Hash(m),求 m 计算困难,除穷举外没有更好的方法;

3)防伪造性:已知 c = Hash(m),求 n 满足 Hash(n) = c 计算困难;

4)初值敏感性:c = Hash(m)中 c 的每一 bit 都与 m 的每一 bit 相关,任意改变 m 中的 1bit,都会对 c 产生明显的影响。

*)基金项目:国家自然科学基金(批准号:60573047),重庆市教委科技项目(KJ070503),重庆邮电大学自然科学基金项目(A2006-41, A2007-26)。王 永 博士,主要研究方向为混沌密码学;廖晓峰 教授,博导,主要研究方向为人工神经网络、混沌理论;杜茂康 副教授,主要研究方向为信息安全。

混沌是非线性确定系统内在的随机性的表现。混沌序列具有对初值和参数极端的敏感性、伪随机性和遍历性等特性，非常适合应用于密码学中。时空混沌除了具有普通混沌的优良特性外，还能在很大程度上防止混沌序列被预测和计算机有限精度的影响。因此，基于时空混沌来构建 Hash 函数不仅可行，而且具有更多的优势。

3 基于时空混沌的单向 Hash 函数构造

3.1 已有算法的不足

在时空混沌的研究工作中，CML 模型因其数字实验的高效率而备受关注。在文献[7]中提出了一种基于 CML 的 Hash 函数的构造，采用的 CML 模型为：

$$x_{n+1}(i)=(1-\varepsilon)f(x_n(i))+\varepsilon[f(x_n(i-1))] \quad (1)$$

其中 $x_n(i)$ 为格子 i 在时刻 n 的状态变量，n 为离散时间坐标，i 为离散空间坐标，$i=1,2,\cdots,N$（N 为 CML 的长度）；$\varepsilon\in(0,1)$ 为耦合系数。周期边界条件为 $x_n(N+i)=x_n(i)$。非线性函数 f 为 logistic 映射，即 $f(x)=\mu x(1-x)$，$x\in(0,1)$，$\mu\in[3.57,4]$。相应的 Hash 函数构造算法如下：

1）待处理消息按对应字节 $C_1,C_2,\cdots,C_N$（C 为消息的 ASCII 码）线性变换为[0,1]范围内的数，得到的数值序列记为 $M_1,M_2,\cdots,M_N$，N 为消息的字节数。计算公式如下：

$$M_i=C_i/256 \quad (2)$$

2）令 $M_1,M_2,\cdots,M_N$ 为 N 个格子的初值，即

$$X_0(i)=M_i \quad (3)$$

取 $\mu=4$，$\varepsilon=0.8$，按照式(1)进行迭代，得到时空混沌序列 N 组：

$$X_n(1),X_n(2),X_n(3),\cdots,X_n(N)$$

3）从迭代结果序列中取出最后一组序列的 $X_R(N)$，$X_{2R}(N)$，$X_{3R}(N)$，且 $R>>N$，将它们线性变换和取整运算映射为两个 40bit 和一个 48bit 的二进制数，合起来作为最后 128bit 的 Hash 值。

该算法具有较好的统计性能，也很好地利用了时空混沌的特性，但存在如下不足：

1）设待处理消息的长度为 N 个字节，即 CML 中格子的个数为 N。由于在 CML 中格子 i 由状态 n 变为状态 $n+1$ 时，需要进行 2 次 logistic 映射运算，因此整个 CML 由状态 n 变为状态 $n+1$ 时，需要进行 $2N$ 次 logisitc 映射运算。产生最终的 Hash 值时，CML 由状态 0 变为状态 $3R(R>>N)$，总共需要进行 6RN 次 logistic 映射运算。若取 $R=N$，则产生 Hash 值时需要计算的 logistic 映射次数与消息长度的平方成正比，所以当处理消息较长时，计算量会显著增加。

2）当待处理消息的长度为一个字节时，CML 模型中仅有一个格子，模型退化为单个的 logistic 映射，不能充分体现 CML 模型的特点。

3.2 耦合映像格子模型分析

本文选用最近耦合方式的 CML 模型来产生 Hash，即

$$x_{n+1}(i)=(1-\varepsilon)f(x_n(i))+\frac{\varepsilon}{2}[f(x_n(i-1))+f(x_n(i+1))] \quad (4)$$

式中各参数意义与式(1)中相同。此 CML 模型的 Lyapunov 指数谱为[13]：

$$LE1=\lim_{n\to\infty}\frac{1}{n}\ln\left|\prod_{m=1}^{n}f'(x_m)\right|$$

$$LE2=LE1+\ln[1-\varepsilon+\varepsilon\cos(2\pi/N)]$$

$$\vdots$$

$$LEn=\begin{cases}LE1+\ln(1-2\varepsilon), & N\text{为偶数}\\ LE1+\ln(1-\varepsilon-\varepsilon\cos(2\pi/N)) & N\text{为奇数}\end{cases} \quad (5)$$

对混沌系统而言，最大 Lyapunov 指数（Largest Lyapunov Exponent：LLE）对系统的混沌特性有至关重要的影响。LLE 越大，系统的混沌特性越好。由式(5)知，此 CML 模型的 LLE（即 LE1）仅与非线性函数 f 有关，而与格子数目 N 和耦合强度 ε 无关。当 f 为 logistic 映射时，LLE 随 μ 增大而增大。由于本文是使用此 CML 模型来构造 Hash 函数，故必须防止模型中各个耦合格子之间出现同步。根据文献[13]的研究结果知，当 LE2 大于 0 可以避免格子之间出现同步，且 LE2 远大于 0 时还能保证格子之间不会出现间歇同步。

CML 模型有多种状态[14,15]，分别为冻结化随机图案状态、图案选择状态、缺陷混沌扩散状态、缺陷湍流状态、图案竞争阵发混沌状态和完全发展混沌状态。为了让 CML 模型具有更好的混沌特性，应该使其处于完全发展的混沌状态。只要耦合强度 ε 不非常小（$\varepsilon>10^{-3}$），则系统可以保持为完全发展的混沌状态[15]。在 $N>5$ 时，由式(5)得 LE2 随 ε 的增大而减小。另外，从避免格子之间出现同步和间歇同步的情况来看，希望 LE2 越大越好，为此又要求 ε 不要过大。

基于以上分析，本文设置 CML 系统的参数值为：$\varepsilon=0.1$，$\mu=3.9999$ 和 $N=16$，对应的周期边界条件为 $x_n(16+i)=x_n(i)$。

3.3 新的 Hash 函数构造算法

3.3.1 算法描述

1）定义 16 个 8-bit 的初始变量，用 16 进制表示为 IV[1…16]=[01,23,45,67,89,AB,CD,EF,FE,DC,BA,98,76,54,32,10]。按照式(6)把它们线性变换为[0,1]内的数，并作为 CML 中对应格子的初始状态值。

$$x_0(i)=IV[i]/256,\ i=1,2,\cdots,16 \quad (6)$$

2）待处理消息按对应字节 $C_1C_2\cdots C_N$（C 为消息的 ASCII 码）线性变换为[0,1]范围内的数，得到的数值序列记为 $M_1M_2\cdots M_N$，N 为消息的字节数。计算式如下：

$$M_i=(C_i+0.5)/256,\ i=1,2,\cdots,N \quad (7)$$

3）对 CML 模型进行如下 $2N$ 步操作：

第 $1-N$ 步：设 $a\in[1,N]$ 为其中某一步骤。在步骤 a 中，按照式(8)更改 CML 中第 1 个格子的状态值，然后将 CML 迭代 K 次，即将 CML 由状态 $(a-1)K$ 转变为状态 aK。

$$x_{a-1}(1)=0.2x_{a-1}(1)+0.8M_a \quad (8)$$

第 $N+1-2N$ 步：设 $b\in[N+1,2N]$ 为其中某一步骤。在步骤 b 中，按照式(9)更改 CML 中第 1 个格子的状态值，然后将 CML 迭代 K 次，即将 CML 由状态 $(b-1)K$ 转变为状态 bK。

$$x_{b-1}(1)=0.2x_{b-1}(1)+0.8M_{2N-b+1} \quad (9)$$

4）最终 CML 的状态为 $2NK$。将 CML 中各格子的状态值转变为二进制形式，从每个格子中抽取 8 比特（小数点后第 9～16 位的数）合并在一起构成 128 位的 Hash 值。

3.3.2 确定每步的迭代次数 K

本文根据 χ^2 检测来确定算法 $2N$ 步操作中的迭代次数 K[16]，具体如下：

1）将区间[0,1]划分为 m 个相等的子区间，表示为 $[(d/m),(d+1)/m]$，$d=0,1,\cdots,m-1$；

2）随机设置 CML 的一个初始状态，令其为 $x_0(i)$，$i=1,2,\cdots,16$。将 CML 迭代 K 次，相应的状态变量表示为 $x_K(i)$，

$i=1,2,\cdots,16$。然后把微小变量 $\Delta\alpha$ 加到 $x_0(1)$ 上，将 CML 再迭代 K 次，相应的状态变量表示为 $x'_K(i)$，$i=1,2,\cdots,16$。按照同样的处理方式，计算出 S 组 $x_K(i)$ 和 $x'_K(i)$。

3)对 CML 中的每个格子(即 i 分别为 $1,2,\cdots,16$)建立一个 $m\times m$ 的频率表。表中单元格的值 n_{ef}(e,f 分别为表的行标和列标)为满足条件 $(e/m)<x_K(i)<(e+1)/(m)$ 和 $(f/m)<x_K'(i)<(f+1)/(m)$ 的数对 $(x_K(i),x_K'(i))$ 的个数。按照公式(10)计算 CML 中每个格子的 χ^2 值。

$$\chi^2=N\left(\sum_{e=1}^{m}\sum_{f=1}^{m}\frac{n_{ef}}{\sum_{f=1}^{m}n_{ef}\sum_{e=1}^{m}n_{ef}}-1\right) \tag{10}$$

4)取各格子中最大的 χ^2 值作为 CML 的 χ^2 值。

本文中取 $m=11$，$S=1000$，$\Delta\alpha=0.8/256$，得到 CML 的 χ^2 值与迭代次数 K 之间的关系如图 1 所示。当 χ^2 值小于 5%对应的临界值时认为两个变量之间无关。此处，自由度为 $(m-1)(m-1)=100$，查 χ^2 表得在 5%临界点时 χ^2_{100} 为124.3。结合图 1 知，当 $K>35$ 时，能保证两个迭代后的 CML 状态完全无关。考虑一定的富余，本文取 $K=40$。

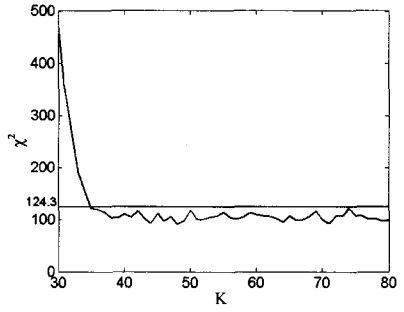

图 1　迭代次数 K 与 χ^2 值的关系

4　性能分析

4.1　文本仿真

取初始文本为“In BECOMING AMERICA, JonButler synthesizes a generation of scholarship to produce a detailed exploration of the maturation of colonial North America after 1680. Despite its rural character and rudimentary technology, Butler asserts that eighteenth-century America was a modem place with a distinctive society”。文本 1 中把初始文本的首字母 I 改为 i，文本 2 中把初始文本内的 1680 改为 1681，文本 3 中把初始文本内的 Despite 写成 Despit，文本 4 中把初始文本内的 character 写成 characters。计算这些文本的 Hash 值，用十六进制数表示如下：

初始文本：E53937BDEA6FAAC1AED25A6845D18451

文本 1：49640BEECE8A0AFFA91E5B7D28644615

文本 2：A5EFD2DC9E9E4007F556018BE6B657CB

文本 3：25E4B00659FB1C0B2F9E9E8EA8F88FAD

文本 4：E0F62FDC6658B365FBFAACFACAC54E77

从计算结果看算法的单向 Hash 性能很好，具有高度的初值敏感性。

4.2　混乱与扩散性质统计分析

混乱与扩散是设计加密算法的两个重要标准，它们对设计 Hash 函数仍然有效。由于 Hash 函数的结果为二进制串形式，每 bit 的取值仅为 0 或 1，因此理想情况下的扩散性表现为初值微小的变化都会引起 Hash 值中每 bit 以 50%的概率变化。在对 Hash 函数进行统计分析时，常考察下列指标：

平均变化 bit 数：$\bar{B}=\frac{1}{N}\sum_{1}^{N}B_i$

平均变化概率：$P=(\bar{B}/128)\times100\%$

B 的均方差：$\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\bar{B})^2}$

P 的均方差：$\Delta P=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i/128-P)^2}\times100\%$

其中 N 为统计总次数，B_i 为第 i 次测试时 Hash 值变化的 bit 数。现按如下方式进行混乱与扩散测试：随机地选取一段明文，计算其 Hash 值。然后随机更改明文中 1 个 bit 的值，计算其 Hash 值，比较两次的 Hash 值得到 B_i。重复上述过程 1024 次，得到置乱数的分布，如图 2 所示。

由图 2 知，在 $N=1024$ 次测试中，明文 1bit 变化引起 Hash 值(128bit)发生变化的 bit 数位于 47 和 81bit 之间，平均 bit 变化数为 63.92，非常接近理想状况下的 64bit 变化数。从图 2 还可看出，B_i 主要集中在 60～70bit 之间，即紧靠在理想值 64bit 附近，这表明算法对明文的置乱能力强而稳定。

另外，再做 $N=256,512,1024,2048$ 次测试，得到明文 1bit 变化时的各项统计值，如表 1 所示。

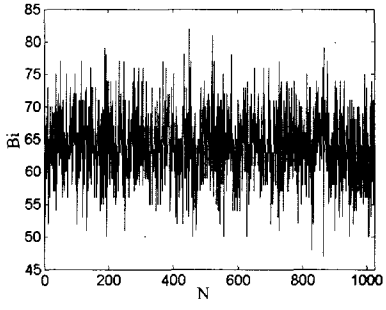

图 2　置乱数分布

表 1　N 次测试的各项指标

N=	256	512	1024	2048	总平均
$\bar{B}$	64.29	64.21	63.92	64.13	64.13
ΔB	5.691	5.809	5.727	5.639	5.716
P(%)	50.23	50.16	49.93	50.11	50.11
$\Delta P(\%)$	4.446	4.538	4.472	4.405	4.465
B_{max}	81	81	82	81	81.25
B_{min}	51	45	47	45	47

由表 1 知，本算法的 B_i 和 P 都非常接近理想状况下的 64bit 和 50%的变化概率。算法很充分和均匀地利用了密文空间，对于明文的任何细微变化，密文从统计上看，在密文空间中都接近等密度的均匀分布，因而攻击者从中得不到任何密文分布的有用信息。另外 $\Delta B,\Delta P$ 为反映 Hash 混乱与扩散稳定性的指标，它们越接近 0 稳定性就越好。本文中算法的 Δ 均很小，能有效保证混乱与扩散能力的稳定。

4.3　碰撞性分析

碰撞是指虽然消息不相同但其 Hash 值却相同，即多对一映射。本文采用文献[5]中的方法进行算法的碰撞性测试，具体如下：随机选择一段明文，将其 Hash 值保存为 ASCII 符的形式。然后随机改变明文中 1 个 bit 的值，将其 Hash 值也保存为 ASCII 符的形式。对两个 Hash 值进行比较，计算它们在相同位置上 ASCII 符相同的个数，并按照式(11)计算两者之间的绝对距离。

$$d=\sum_{i=1}^{N}|t(e_i)-t(e_i')| \tag{11}$$

式中 e_i 和 e_i' 分别为两个 Hash 值中的第 i 个 ASCII 符，函数 t ()表示将 ASCII 符转换为对应的数值。重复上述过程 2048 次，得到的最大、最小和平均绝对距离如表 2 所示。同时这 2048 个 Hash 值在相同位置上具有相同 ASCII 符的个数分布，如图 3 所示。从图中可以看出，在相同位置上有相同 ASCII 符的个数最多为 2 个，说明算法的碰撞率很低。

表 2　绝对距离

	最大值	最小值	平均	每字符平均
绝对距离	2141	675	1386	86.625

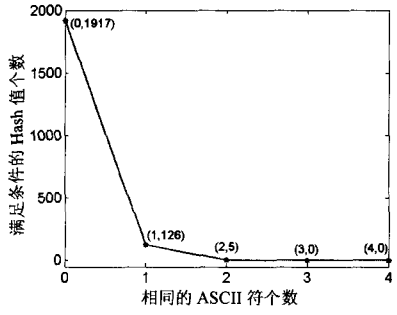

图 3　在相同位置上具有相同 ASCII 符的 Hash 值个数分布

4.4　灵活性

本算法具有较好的灵活性，能满足多种情况的需要。比如，若需要将 Hash 值的位数由 128 位增加为 160 位或 256 位时，只需在算法中简单地增加格子个数或增加从每个格子中抽取的 bit 数即可。另外，当需要使用有密钥的 Hash 函数时，只需将算法中每个格子的初始值更改为密钥即可。

4.5　对比分析

在本文的 Hash 算法中，由于 CML 有 16 个格子，因此当 CML 由状态 n 变为状态 $n+1$ 时需要进行 48 次 logistic 映射运算。当输入文本包含 N 个字符时，由于最后 CML 的状态为 $80N$，因此需要计算的 logistic 映射次数为 $3840N$。这表明产生 Hash 值时需要计算的 logistic 映射次数与消息长度之间为线性关系，能有效避免消息较长时计算量急剧增加的问题。在不同消息长度下，本文算法与文献[7]中算法在配置为奔腾Ⅳ2.8GHz，256M 内存的个人电脑上产生 Hash 值的时间分别如图 4 和图 5 所示。从中可以看出本文算法的计算效率更高，特别是当消息长度较大时优势更为明显。

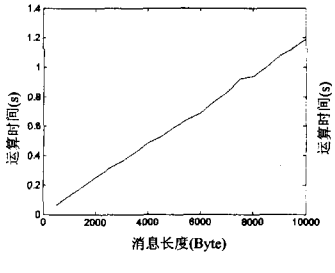

图 4　不同消息长度时，本文算法产生 Hash 值的时间

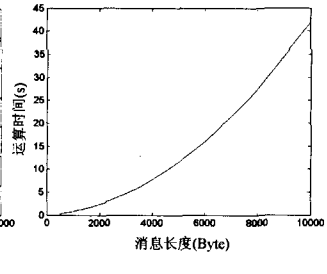

图 5　不同消息长度时，文献[7]算法产生 Hash 值的时间

另外，当消息长度仅为 1 个字符时，本文算法仍然可以保持时空混沌的优良特性，不会退化为单个的 logistic 映射。

结束语　本文根据 Lyapunov 指数谱和 CML 的状态特性来确定时空混沌中的各参数值，并在此基础上提出了一种新的基于时空混沌的单向 Hash 函数构造方案。理论分析和仿真试验表明：1)算法能有效克服混沌序列被预测和计算机有限精度的影响。2)无论消息长度如何，该构造算法都很好地保持了时空混沌的特性，具有很好的单向性和安全性。3)与已有的基于时空混沌的 Hash 函数构造方案相比，本文算法能有效减少计算 Hash 值的时间，具有更高的效率。

参 考 文 献

[1] Menezes A , Oorschot P , Vanstone S . Handbook of Applied Cryptography[M]. CRC Press,1996

[2] 李红达,冯登国. 复合离散混沌动力系统与散列函数. 计算机学报[J],2003,4 (26):21-26

[3] Xun Yi. Hash Function Based on Chaotic Tent Maps[J]. IEEE Transactions on Circuits and Systems-Ⅱ,2006,52 (6):354-357

[4] 王小敏,张家树,张文芳. 基于广义混沌映射切换的单向 Hash 函数构造[J]. 物理学报,2003,52 (11):2737-2742

[5] Wong Kwok-Wo. A combined chaotic cryptographic and hashing scheme[J]. Physics Letters A,2003,307 (5/6):292-298

[6] Xiao Di,Liao Xiaofeng,Deng Shaojiang. One-way Hash function construction based on the chaotic map with changeable-parameter[J]. Chaos Solitons & Fractals,2005,24 (1):65-71

[7] 张瀚,王秀峰,李朝晖,等. 基于时空混沌系统的单向 Hash 函数构造[J]. 物理学报,2005,54 (9):4006-4011

[8] 刘军宁,谢杰成,王普. 基于混沌映射的单向 Hash 函数构造[J]. 清华大学学报:自然科学版,2007,40 (7):55-58

[9] Short K M. Signal extraction from chaotic communications[J]. International Journal of Bifurcation and Chaos, 1997, 7 (7): 1579-1597

[10] Short K M. Steps toward to unmasking secure communications [J]. International Journal of Bifurcation and Chaos,1994,4 (4): 959-977

[11] Li Ping,Li Zhong,Halang W A,et al. A multiple pseudorandom-bit generator based on a spatiotemporal chatic map[J]. Physics Letters A,2006,349 (6):467-473

[12] Kou Weidong. Network security and standards [M]. Boston: Kluwer Academic Pulishers,1997

[13] Ding Mingzhou, Yang Weiming. Stability of synchronous chaos and on-off intermittency in coupled map lattices[J]. Physical Review E,1997,56 (10):4009-4016

[14] 杨维明. 时空混沌和耦合映像格子[M]. 上海:上海科技教育出版社,1994

[15] Kaneko K . Pattern dynamics in spatiotemporal chaos : pattern selection, diffusion of defect and pattern competition intermittency[J]. Physica D,1989,34 (1/2):1-41

[16] Habutsu T,Nishio Y,Sasase I,et al. A secret key cryptosystem by iterating a chaotic map[C]// Eurocrypt'91. Brighton, U K, 1991