基于分段线性混沌映射的算术编码与加密*

王小龙，赵庶旭
（兰州交通大学 电子与信息工程学院，兰州 730070）

摘 要：为了提高数据的编解码速率和传输安全，根据迭代斜帐篷映射的反函数可以作为算术编码，提出了基于分段线性混沌映射的算术编码与加密方案。方案将加密嵌入编码过程中，解决了现存方案中编码与加密独立分离的问题，实现了同步编码与加密。结果表明，算法提高了编码效率和密文的不可预测性，并且可以有效抵御选择性明文攻击、短明文攻击和蛮力搜索攻击。

关键词：分段线性混沌映射；算术编码；同步；加密
中图分类号：TP309.7 **文献标志码**：A **文章编号**：1001-3695(2014)05-1481-03
doi:10.3969/j.issn.1001-3695.2014.05.047

Arithmetic coding and encrytion based on piecewise linear chaotic maps

WANG Xiao-long, ZHAO Shu-xu
(*School of Electronic & Information Engineering, Lanzhou Jiaotong University, Lanzhou* 730070, *China*)

Abstract: In order to improve the data coding rate and transmission security, according to the inverse function of iteration skew tent map, which could be used as arithmetic coding, this paper propsed a compression and encryption scheme based on the piecewise linear chaotic map. It solved the separate problems of encoding and encryption in the existing program. The simulation results indicate that the algorithm improves the coding efficiency and unpredictability of ciphertext. At the same time, it can effectively resist the attack of selective plaintext, short plaintext attack and brute force search attack.

Key words: piecewise linear chaotic map; arithmetic code; synchronization; encryption

0 引言

近些年，由于机密数据在公共网络中传输数据量的不断增加，使得数据压缩与加密的效率和安全成为更多人关心的问题。传统的数据压缩与加密是分离的，这就使得传统数据编码与加密很难满足当今数据量增大、并且要求在不降低编码效率的同时提高编解码的速率和数据传输安全。所以将编码与加密结合起来实现同步编码与加密是值得研究的。

目前常用的有两种方式：a)将编码嵌入到加密体系中；b)编码的同时引入密钥。针对方法 b)已经有很多种方案被提了出来，但是大多数方案在效率和安全上面都有很大的缺陷。文献[1]基于区间分割的算术编码方案容易受已知明文攻击；文献[2]中的加密算术编码方案已经被破解；文献[3]中的随机算术编码方案虽然没有被密码分析方法破解，但是它在效率上低于传统编码与加密方案[4]；文献[4]中指出多霍夫曼表的编码方案已被密码分析方法破解。近些年，很多基于混沌系统的信源编码方案被提出，尤其是关于混沌在算术编码方案中的研究更多，但是这些方案并没有研究混沌与算术编码之间的关系，只是单纯地把混沌系统作为伪随机序列发生器来使用。此外基于混沌系统的自适应编码方案的缺点在文献[5]中被发现，文献[6]对其进行了改进，但是改进后的方案仍然易受到选择性明文攻击。

混沌是一种非线性动力学规律控制的行为，表现为对初始值和系统参数的敏感性、白噪声的统计特性和混沌序列的遍历特性，其吸引子的维数有十分复杂的分形结构，具有不可预测性。随着混沌加密的深入研究，基于混沌的很多有关信源编码特别是算术编码和霍夫曼编码的编码与加密方案被提了出来[5~8]。但是这些方案并没有实现编码与加密的同步，仍然是将编码与加密分开来对待，没有取得在编码和加密同步方面的进展。

1 基于斜帐篷映射的算术编码

斜帐篷映射的一般定义为

$$f(x)=\begin{cases} x/p & x\in[0,p) \\ (1-x)/(1-p) & x\in[p,1] \end{cases} \tag{1}$$

该映射是通过参数 p 来得到斜帐篷映射的，当 $p=0.5$ 时，该映射为标准帐篷映射。

假设符号“0”出现的概率为 0.6，而符号”1”出现的概率为 0.4，其中 p 为区间分割点，当 $p=0.6$ 时，区间[0,1]被分割为[0,0.6)和[0.6,1]两个区间，其分别代表 0 和 1。斜帐篷映射的反函数可以作为算术编码，即

$$f^{-1}(I)=\begin{cases} p\times I & s=\text{“0”} \\ 1-(1-p)\times I & s=\text{“1”} \end{cases} \tag{2}$$

其中：I 表示符号落入的区间，初始区间 $I=[0,1]$。假设要求编码的序列 $S=1010111$，则第一个字符“1”将区间[0,1]映射到[0.6,1]，第二个字符为“0”，则将区间[0.6,1]映射到[0.36,0.6]，第三个字符为“1”，则将区间[0.36,0.6]映射到[0.76,0.856]。同理将剩余的字符逐个编码，得到最终的编码区间，然后在最终的区间中选择任意的数进行二值化并将其作为最终的压缩序列。

收稿日期：2013-07-05；**修回日期**：2013-08-14 **基金项目**：国家自然科学基金资助项目(60962004)；校科技支撑项目(zc2013012)

作者简介：王小龙(1989-)，男，甘肃定西人，硕士研究生，主要研究方向为信息融合处理(small25300@163.com)；赵庶旭(1976-)，男，山东青岛人，教授，博士，主要研究方向为自律分散系统与智能信息处理.

如果是多个字符序列进行编码,可以将斜帐篷映射进行扩展来完成对序列的编码。在文献[9]中对其进行了构造,其表达式为

$$f(x)=\begin{cases} x/P_1 & x\in I_1 \\ (x-P_1)/P_2 & x\in I_2 \\ \vdots & \\ (x-\sum_{i=1}^{n-1}P_i)/P_n & x\in I_n \end{cases} \tag{3}$$

其中:$P_i(i=1,\cdots,n)$表示序列中每种符号出现的概率,而区间 I_i 表示累积概率区间。

$$\sum_{i=1}^{n}P_i=1 \tag{4}$$

$$I_1=[0,P_1] \tag{5}$$

$$I_i=[\sum_{j=1}^{i-1}P_j,\sum_{j=1}^{i}P_j] \qquad i=2,3,\cdots,n \tag{6}$$

其映射图如图 1 所示。

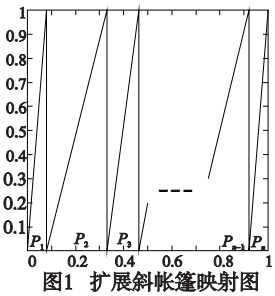

图1 扩展斜帐篷映射图

由式(3)可知,$f(x)$的反函数为

$$f^{-1}(x)=\begin{cases} P_1x \\ P_2x+P_1 \\ \vdots \\ P_nx+\sum_{i=1}^{n-1}P_i \end{cases} \tag{7}$$

根据信源熵理论,一般信源编码符号 s_i 的自信息量为 $-\log_2(P(s_i))$,相应序列的平均信息量为

$$H=-\sum_{i=1}^{n}P_i\log_2(P_i) \tag{8}$$

而序列 $s=\{s_i,i=1\cdots M\,|\,s_i\in S\}$,其中 S 表示序列中出现的符号种类的集合。而 s 中每个符号出现的概率可表示为

$$P_n=\frac{1}{M}\mathrm{card}\{s_i\,|\,s_i=S_n\} \tag{9}$$

其中:$S_n\in S,n=1\cdots N$。

由于式(7)与式(3)互为反函数,所以其区间也是反向对应的,可得

$$I^i=f^{-1}(x)(I^{i+1}) \tag{10}$$

$$\mathrm{size}(I^i)=P_i\mathrm{size}(I^{i+1}) \tag{11}$$

则

$$\mathrm{size}(I^1)=\Pi_{i=1}^{M}P(s_i\in S)=\Pi_{n=1}^{N}(P_n)^{\mathrm{card}\{s_i\,|\,s_i=S_n\}}=\Pi_{n=1}^{N}(P_n)^{P_nM} \tag{12}$$

所以得

$$-\log_2(\mathrm{size}(I^1))=-\sum_{n=1}^{N}P_nM\log_2(P_n)=M\times H \tag{13}$$

最后压缩编码后序列的最佳编码长度为 $M\times H$[9]。

2 算术编码与加密方案

编码与加密方案框图如图 2 所示,在此方案中,编码的同时将会产生加密所需的一个参数密钥。而编码后的序列将由另一个混沌映射迭代产生的伪随机序列进行加密,伪随机序列的产生是由之前编码所生成的参数密钥和混沌初始值共同来决定的。这种方法提高了明文、编码与加密之间的相关性,使得编码与加密构成一个统一的整体。

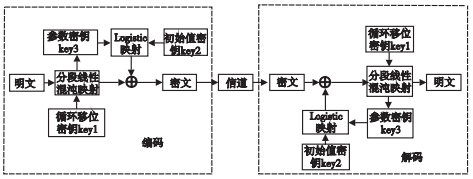

图2 算术编码与加密框图

2.1 混沌序列发生器

Logistic 映射是一维的非线性系统,且能出现极复杂的混沌特性,由于其表达式简单、随机性能好,被广泛地应用于混沌加密和保密通信的各个领域。Logistic 映射的定义为[10]

$$X_{n+1}=\mu X_n(1-X_n)\quad \mu\in(0,4],X_n\in[0,1] \tag{14}$$

当参数 $\mu\in[3.56994\cdots,4]$ 时,Logistic 映射处于混沌状态。参数 μ 作为密钥之一,它的产生是基于循环移位密钥和所要编码字符序列的第一个字符,在密钥模式下编码时首先要对明文的累积概率区间段进行循环移位。如果对长度为 L 的明文在密钥模式下进行编码,则对应的累积概率区间段要变换 L 次,根据这 L 次中明文第一字符概率所在的区间段来计算 μ 值,如图 3 所示。

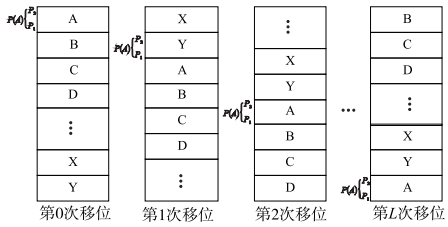

图3 概率区间移位变换图

假设某一序列的首字符为 A,第 0 次循环移位后对应的 μ 值记为 μ_0,依次往后第 L 次循环移位后的 μ 值记为 μ_L,则

$$\begin{cases} \mu_0=(P_{01}+P_{02})/2 \\ \mu_1=(P_{11}+P_{12})/2 \\ \mu_2=(P_{21}+P_{22})/2 \\ \vdots \\ \mu_L=(P_{L1}+P_{L2})/2 \end{cases} \tag{15}$$

$$\mu=3.57+\frac{1}{L}\sum_{i=0}^{L}\mu_i \tag{16}$$

然后将最终得到的 μ 值作为 Logistic 映射迭代的参数,μ 的值将随着明文序列和循环移位密钥的改变而随之改变。

2.2 循环移位密钥

假设明文序列由 n 种不同字符组成,则能表示循环移位数 $KC\in\{0,1,\cdots,n-1\}$ 的最少比特数为$\lceil\log_2(n)\rceil$位,而循环移位方向 $KS\in\{0,1\}$为 1 位,所以循环移位密钥总的比特位数 $m=\lceil\log_2(n)\rceil+1$。选择密钥存储空间为 K bit,每个加密编码的字符各自对应一个循环移位密钥,即要加密编码的字符数 $L=\lceil K/m\rceil$,则剩余的为非加密编码的字符数。选择的密钥空间越大,用加密编码的明文数就越多,其安全性也就越高。为了验证时编码速度相对较快,选取 $K=256$,则 key1 $=k_1,k_2,\cdots,k_{256}$,随机产生 256 bit 的二进制序列,作为循环移位密钥[11]。

2.3 编码

编码过程如下所述:

a)读取明文序列然后将明文分成 N 个块,其中每个字符按 8 位来读取,前 $N-1$ 个块每个块包含 128 个字符,第 N 个块的字符数为 $k(128\leqslant k\leqslant 256)$,这样使得每个块至少有 1 024

bit,可以阻止强力搜索攻击。

b)对每个块进行概率统计,根据式(3)~(6)来构建斜帐篷映射,然后对每个明文块进行压缩编码。其中每个块的前 L 个字符进行加密编码,剩余的 128 - L 个字符进行非加密编码,然后得到最终的编码区间。在最后的编码区间中选择一个实数值二值化来作为最终的压缩编码序列 0. $t_1t_2t_3\cdots t_r$ ($r\geqslant 128$);再对 $t_1t_2t_3\cdots t_r$ 用 Logistic 映射产生的混沌序列来加密,初始值 x_0 随机产生,而参数 μ 是由字符累积概率区间和明文序列的首字符决定。为了提高混沌序列的随机性,首先迭代 50 次,再迭代$\lceil r/16\rceil$得到$\lceil r/16\rceil$个序列值,然后将序列中的每个值进行二值化处理得到长度为 16 的二进制小数,最终组成所需的加密序列 $x_1x_2x_3\cdots x_r$。所以最后所得的密文序列 $c_1c_2c_3\cdots c_r$ 是由编码序列和加密序列按位异或得到,即 $c_i=t_i\oplus x_i, i=1,2,\cdots,r$。

c)对每个分块编码加密之后要对密钥进行更新,更新方式如下:

$$\begin{cases} KC_1 = KC_1 \oplus \{t_j, t_{j+1}, t_{j+2}, \cdots, t_{j+m-2}\} \\ KS_1 = KS_1 \oplus \{t_{j+m-1}\} \end{cases} \tag{17}$$

其中:$j = r \bmod 128$,如果 $j > r-m$ 则 $j = j-m$;如果 $j=0$,则 $j=m$。将 Logistic 混沌映射的初始值更新为 $x_0 = x_{\lceil r/16\rceil+1}$,根据上述步骤将剩余分块依次编码加密,编码加密完成。

d)由于解码是按照块来进行的,为了能够区分块序列将序列"011111111"插入每个块序列之间。为了保证每个块序列仅有一个标志序列,将所有连续的 7 个"1"后加入一个"0"。

2.4　解码

解码是编码的逆过程,过程如下所述:

a)读取密文序列,找到每个块的标志序列、字符概率和序列首字符。然后根据式(3)~(6)构建非线性混沌映射,扫描密文序列找到标志序列"011111111",则第一个分块密文找到,紧接着扫描块序列看是否有连续的 7 个"1",如果有则去掉其后面的"0"。

b)循环密钥 key1 和序列首字符根据式(15)(16)计算出 Logistic 映射的参数μ,设定初始值 x_0,然后迭代 50 次,再迭代$\lceil r/16\rceil$得到$\lceil r/16\rceil$个序列值,然后得到长度为 16 的二进制小数,最终组成所需的加密序列 $x_1x_2x_3\cdots x_r$,则 $t_i=c_i\oplus x_i, i=1,2,\cdots,r$。将二进制序列 $t_1t_2t_3\cdots t_r$ 转换为实数值 0. $t_1t_2t_3\cdots t_r$ ($r\geqslant 128$),然后根据循环密钥 KC_i 和 KS_i 对前 L 个字符进行解压,剩余的字符用非加密编码方式解压。

c)更新循环密钥 KC_i 和 KS_i,然后用 $x_0 = x_{\lceil r/16\rceil+1}$来更新迭代初始密钥,最后用更新好的密钥解码剩余序列,直至结束。

3　结果与分析

编码的性能和安全采用标准测试文件进行了测试,测试结果如图 4 所示。

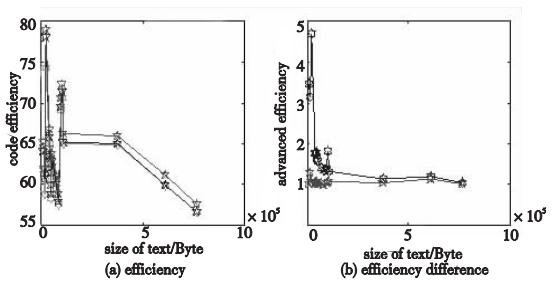

图4　编码性能图

3.1　压缩性能

压缩比是用来衡量编码算法的一个性能指标,其定义为

$$R = \frac{L_{\text{coded}}}{L_{\text{uncoded}}} \times 100\% \tag{18}$$

对 16 个大小不同的标准测试文件按由小到大的顺序进行测试,得到编码效率如图 4 所示。可以明显看到,随着文件增大,压缩比也相对减小,当测试文件足够大时,编码效率特别接近理论值。如图 4(b)所示,相对于标准算术编码,提高了 1% 左右,而相对于理论最佳编码效率提高了 1% ~4.5% 左右,文件越小,效率越高。当编码算法固定以后,分块的大小决定了编码的效率,因此,当文件大小大于 3.8×10^5 Byte 时,可以将每个块大小分为 256 Byte 用来提高编码效率。根据以上分析,可以采用自适应的分块方法增加编码效率,随时根据文件的大小来减小或增加分块的长度。

3.2　随机性测试

加密密钥序列的随机性测试使用 SP800-22 进行测试,测试的部分结果如图 5 所示。

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <randomcipher.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
13	9	7	12	12	7	9	16	5	10	0.366918	99/100	Frequency
11	12	11	10	9	17	9	6	8	7	0.474986	96/100	BlockFrequency
10	7	14	7	7	13	16	10	10	6	0.319084	99/100	CumulativeSums
9	14	8	11	6	11	12	6	9	14	0.574903	98/100	CumulativeSums
16	6	13	7	13	8	7	8	11	11	0.366918	97/100	Runs
13	5	12	9	13	13	9	6	11	9	0.574903	100/100	LongestRun
10	12	5	17	15	12	7	8	8	6	0.122325	99/100	Rank
15	15	12	8	7	6	3	15	7	12	0.048716	100/100	FFT
9	9	10	21	8	9	5	9	5	15	0.015598	98/100	NonOverlappingTemplate
10	11	11	13	8	8	7	4	11	17	0.249284	99/100	Serial
8	8	8	13	8	9	13	9	11	13	0.867692	100/100	Serial
11	5	8	5	14	13	14	8	6	16	0.085587	99/100	LinearComplexity

图5 NIST部分测试结果

SP800-22 测试共有 15 项测试,主要以 P-value 的值与参数 α 进行比较,一般 α 在 0.001 ~0.01 之间,本次测试时 α 取值为 0.01。测试产生的加密序列,共测试了 100 组,每组的长度为 1 0000。从图 5 可知 P-value 的值大于 α,且测试的成功率基本在 97% ~100%,因此可以认为加密密钥序列具有足够的随机性[12]。

3.3　安全及敏感性分析

改变明文块的首字符,同时改变循环移位密钥中的一位,然后对比之前未改变时生成的密文序列进行测试。共对块首字符和循环移位密钥改变 50 次,得到密文序列的平均变化率在 47.652% ~49.926% 之间,都接近理想值 50%,部分结果如表 1 所示。

表 1　密文敏感性测试变化率

首字符	第 i 位(改变后)	变化率/%	首字符	第 i 位(改变后)	变化率/%
A	2(0)	48.26	T	3(1)	47.61
F	4(1)	47.92	B	1(1)	49.28
D	5(0)	49.37%			

通过 3.2 节中密钥的随机性测试结果可知,加密密钥具有充分的随机性,因此可以有效抵御已知明文攻击、短明文攻击和选择性明文攻击。通过对明文的敏感性分析可知,由于分块的不同和明文字符的不同使得每个分块的首字符处于不断的变化当中,对蛮力搜索攻击具有很好的抵御效果,因此具有很高的安全性。

4　结束语

本文基于混沌映射的遍历性和复杂性,利用分段线性混沌映射和 Logistic 映射具有较大的参数空间和相空间,将分段线

(下转第 1487 页)

机选取 τ 个元组,将 $\tau/2$ 个元组乘以 1.01,另外 $\tau/2$ 个元组乘以 0.99)和子集增加(复制随机选取的 τ 个元组)攻击,仿真结果分别如表 1 ~ 3 所示。由于采用重复嵌入水印的方式,水印算法对于上述攻击具有较强的鲁棒性。

表 1　子集删除攻击所产生的水印提取率

元组选取比例/%	≤40	50	60	70	80	85	90
水印提取率/%	100	100	100	100	100	69	51

表 2　子集修改攻击所产生的水印提取率

元组选取比例/%	≤40	50	60	70	80	90	100
水印提取率/%	100	100	100	100	100	100	100

表 3　子集增加攻击所产生的水印提取率

元组选取比例/%	≤40	50	60	70	80	90	100
水印提取率/%	100	100	100	100	100	100	100

3.3　不可见性分析

嵌入水印对数据库数据各属性列的整体影响如表 4 所示。可以看出由于嵌入水印所引入的误差较小,水印算法具有良好的不可见性。

表 4　嵌入水印对数据均值和方差的影响

二进制比特水印序列	属性名	均值改变比例/%	方差改变比例/%
0 和 1 的个数均等	A_1	1.228	0.699
	A_2	0.279	1.201
1 的数目多	A_1	1.368	0.200
	A_2	0.582	1.801
0 的数目多	A_1	1.297	0.798
	A_2	0.022	1.703
全 1 序列	A_1	1.297	0.689
	A_2	0.031	1.202
全 0 序列	A_1	1.072	1.159
	A_2	0.041	0.897

4　结束语

本文设计了一种基于 PSO-BF 优化算法的关系数据库水印算法,利用混沌映射、单向 hash 函数和密钥等信息安全技术提高算法的鲁棒性,利用 PSO-BF 优化算法提高算法的不可见性,理论分析和实验结果证明该算法能够有效地解决数据库的版权保护问题。如何将优化算法应用于非数值型数据的水印嵌入和提取,将是今后继续研究的方向。

参考文献:

[1] MAO Li, FAN Yang-yu, WANG Hui-qin, *et al*. Fractal and neural networks based watermark identification [J]. Multimedia Tools and Applications, 2011,52(1):201-219.

[2] 周飞,赵怀勋. 基于混沌的 DCT 域关系数据库水印算法[J]. 计算机应用研究,2012,29(2):786-788.

[3] PASSINO K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. IEEE Control Systems Magazine, 2002, 22(3):52-67.

[4] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. Piscataway: IEEE Service Center, 1995:1942-1948.

[5] DAS S, BISWAS A, DASGUPTA S. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications [C]// Foundations of Compututational Intelligence. Berlin: Springer, 2009: 23-55.

[6] SHEHAB M, BERTINO E, GHAFOOR A. Watermarking relational databases using optimization based techniques [J]. IEEE Trans on Knowledge and Data Engineering, 2008,20(1):116-129.

[7] SION R, ATALLAH M, PRABHAKAR S. Rights protection for relational data [J]. IEEE Trans on Knowledge and Data Engineering, 2004,16(12):1509-1525.

(上接第 1483 页)性混沌映射用于算术编码,提出了基于混沌映射的同步算术编码与加密方案。本算法解决了现存方案中编码和加密独立分开的问题,将加密的过程嵌入编码的过程中,实现了同步编码与加密。整个加密过程十分复杂,提高了密文的不可预测性,可以抵御短明文攻击和蛮力搜索攻击。仿真结果表明,产生的密钥具有充分的随机性且密钥与密文具有很强的敏感性。

参考文献:

[1] WEN Jiang-tao, KIM H, VILLASENOR J. Binary arithmetic coding with key-based interval splitting[J]. IEEE Signal Process, 2006,13(2):69-72.

[2] KIM H, WEN Jiang-tao, VILLASENOR J. Secure arithmetic coding [J]. IEEE Trans on Signal Process, 2007,55(5):2263-2272.

[3] GRANGETTO M, MAGLI E, OLMO G. Multimedia selective encryption by means of randomized arithmetic coding[J]. IEEE Trans on Multimedia, 2006,8(5):905-917.

[4] JAKIMOSKI G, SUBBALAKSHMI K. Cryptanalysis of some multimedia encryption schemes[J]. IEEE Trans on Multimedia, 2008,10 (3):330-338.

[5] BOSE R, PATHAK S. A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system[J]. IEEE Trans on Circuits System, 2006,53(4):848-857.

[6] ZHOU Jian-jiang, AU O C. Comments on a novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system [J]. IEEE Trans on Circuits System, 2008,55 (10):3368-3369.

[7] MI Bo, LIAO Xiao-feng, CHEN Yong. A novel chaotic encryption scheme based on arithmetic coding [J]. Chaos Solitons Fractals, 2008,38(5):1523-1531.

[8] LI Heng-jian, ZHANG Jia-shu. A secure and efficient entropy coding based on arithmetic coding[J]. Communications in Nonlinear Science and Numerical Simulation, 2009,14(12): 4304-4318.

[9] LUCA M B, SERBANSCU A, AZOU S, *et al*. A new compression method using a chaotic symbolic approach[C]//Proc of IEEE Communications Conference. 2004:3-5.

[10] 王飞,王海春. 混沌加密技术在 WSN 中实现的分析[J]. 计算机科学,2010,37(7A):136-138.

[11] WONG K W, LIN Qiu-zhen, CHEN Jian-yong. Simultaneous arithmetic coding and encryption using chaotic maps[J]. IEEE Trans on Circuits and Systems,2008,57(2):146-150.

[12] RUKHIN A, SOTO J, NECHVATAL J, *et al*. A statistical test suite for random and pseudorandom number generators for cryptographic applications [EB/OL]. (2010-04-08). http://csrc. nist. gov/groups/ST/toolkit/rng/documentation_software. html.