基于 PSO-BF 优化算法的关系数据库水印算法*

毛 力[1,2]，樊养余[1]，王慧琴[2]，王 可[2]
(1. 西北工业大学 电子信息学院，西安 710072；2. 西安建筑科技大学 信息与控制工程学院，西安 710055)

摘 要：借鉴群体智能优化算法，提出了一种基于粒子群优化的细菌觅食算法(PSO-BF)的关系数据库水印算法。运用混沌映射对水印序列加密生成水印信号，采用单向 hash 函数加密关系数据库中元组的次序，利用 PSO-BF 优化算法解决将水印信号的嵌入过程抽象为约束条件下的优化问题，阈值解码和择多表决法用于提取水印信号，并通过混沌映射解密为表征版权的水印序列。实验表明，该算法对元组删除、修改和添加等攻击具有较强的鲁棒性和良好的不可见性。

关键词：关系数据库水印；细菌觅食优化算法；粒子群优化算法；PSO-BF；混沌映射
中图分类号：TP309.2 **文献标志码**：A **文章编号**：1001-3695(2014)05-1484-04
doi:10.3969/j.issn.1001-3695.2014.05.048

Relational database watermarking algorithm based on PSO-BF

MAO Li[1,2], FAN Yang-yu[1], WANG Hui-qin[2], WANG Ke[2]
(1. *School of Electronics & Information, Northwestern Polytechnical University, Xi'an* 710072, *China*; 2. *School of Information & Control Engineering, Xi'an University of Architecture & Technology, Xi'an* 710055, *China*)

Abstract: This paper presented a relational database watermarking algorithm based on PSO-BF by adopting swarm intelligence optimization algorithm for reference. The application of chaotic mapping resulted in watermark. The use of one-way hash function secured order of the tuples. It formulated the watermarking of relational databases as a constrained optimization problem, and used PSO-BF algorithm to solve the optimization problem and to handle the constraints. It decoded watermark bits by using threshold-evaluator and majority-voter, then decrypted them to derive the actual watermark sequence by utilizing chaotic mapping. The experimental results show that this algorithm has good robustness to the attacks of tuple deletion, alteration and insertion, and possesses good invisibility.

Key words: relational database watermarking; bacteria foraging optimization; particle swarm optimization; PSO-BF; chaotic mapping

0 引言

随着大数据时代的来临，数据安全等围绕大数据商业价值的利用成为学术界研究的热点。数据的价值不仅仅局限于它的基本用途，更多在于它的二次甚至多次利用。确保数据库的数据安全性是一个复杂的问题，涉及技术、标准和法律等众多因素，目前为止尚不能找到一种解决方案来应对各种数据侵权行为。数字水印技术为数据的安全存储和传输开辟了一条全新的途径，该技术通过在数据中嵌入具有某种特征的水印信息，以达到版权保护的目的。水印是向被保护的数据中嵌入某些能证明版权归属或跟踪侵权行为的信息，可以是数据拥有者的标志、有意义的符号序列等[1]。目前大部分的数据都是存储在关系数据库中，本文提出了一种基于粒子群优化的细菌觅食算法(particle swarm optimization-bacteria foraging, PSO-BF)的关系数据库水印算法。该算法将混沌理论、加解密技术和 PSO-BF 优化算法相结合，提高了水印的隐蔽性，增强了算法的鲁棒性。水印的提取过程只需要秘密参数，不需要原始数据的参与，属于盲水印算法。

1 相关算法介绍

1.1 基于 Logistic 混沌映射的加密算法

混沌序列既非周期性又不收敛，是一种具有时间遍历性和连续宽带频谱的随机过程，适合用于加密水印信号。Logistic 映射是典型的、也是应用最广泛的混沌模型[2]。

一维 Logistic 映射定义为

$$X_{n+1}=\mu\times X_n(1-X_n) \tag{1}$$

式中，$X_n\in(0,1)$，当 $3.5699456<\mu\leqslant 4$ 时，Logistic 映射处于混沌状态，迭代值是一种伪随机分布的状态。本文采用的方法是将作为版权标记的二进制序列进行 Logistic 混沌映射，将加密后的数字序列作为水印信号嵌入数据库中，这样可以加强水印的保密性。

1.2 基于粒子群优化的细菌觅食算法(PSO-BF)

细菌觅食优化算法(bacteria foraging optimization, BFO)是由 Passino[3]在 2002 年提出的一种新的生物启发优化算法。BFO 算法针对具体问题的求解过程为：产生初始解群体、计算

收稿日期：2013-07-06；**修回日期**：2013-07-29 **基金项目**：国家“863”计划资助项目(2007AA01Z324)；国家教育部高校回国人员扶持基金资助项目(K05055)；陕西省教育厅产业化项目(2011JG12)；陕西省教育厅专项科研资助项目(11JK1048)

作者简介：毛力(1980-)，男，山东博兴人，讲师，博士研究生，主要研究方向为数字水印技术、三维重建技术(morley220@qq.com)；樊养余(1960-)，男，教授，博导，博士，主要研究方向为数字信号处理、信息安全；王慧琴(1970-)，女，教授，博导，博士，主要研究方向为数字图像处理、多媒体技术；王可(1981-)，男，讲师，博士研究生，主要研究方向为数据库系统.

适应度函数的值、利用群体的相互影响和作用机制进行迭代优化,通过循环执行趋化(chemotaxis)、繁殖(reproduction)和迁移(elimination and dispersal)三个主要算子来获得最优解或准最优解。标准的 BFO 算法由于细菌趋化行为的随机性,导致趋化速度较慢,优化解精度不高。为了改进 BFO 算法的性能,研究人员提出了各种方案,包括对算法参数的调整以及融合其他智能算法。粒子群优化算法(particle swarm optimization, PSO)借鉴鸟类行为中的种群学习和自我学习概念[4],使得算法速度很快,是一种全局优化进化算法。在 D 维解空间中,粒子群中每个粒子的位置 X 表示所求问题的候选解,粒子在执行算法搜索时,速度决定着粒子更新的方向和距离。在 k 代,第 i 个粒子的位置和速度分别记为 $X_i(k)$ 和 $V_i(k)$;$P_i(k)$ 表示粒子位置的历史最优,$P_g(k)$ 表示粒子群位置的历史最优;w 非负数的惯性权重,它使粒子具有扩展收缩空间的能力,有助于提高算法全局寻优能力;α 和 β 为[0,1]非负数的学习因子,它们使粒子随机地向个体最优和群体最优进行自主学习;C_1 和 C_2 为内均匀分布的伪随机数。每一代粒子根据式(2)和(3)更新自己的速度和位置。

$$V_i(k+1)=wV_i(k)+\alpha C_1[P_i(k)-X_i(k)]+\beta C_2[P_g(k)-X_i(k)] \tag{2}$$

$$X_i(k+1)=X_i(k)+V_i(k+1) \tag{3}$$

本文采用一种基于粒子群优化的细菌觅食算法(PSO-BF),借鉴 PSO 算法中粒子位置的历史最优和粒子群位置的历史最优思想,将 BF 算法中的趋化算子进行改进。在趋化算子执行时,将适应度优于平均值的细菌,通过追踪最优细菌进行搜索;反之,通过追踪细菌群中心位置进行搜索;若上述两种方式都失效,则采用标准 BF 算法的随机搜索方式。

PSO-BF 算法的主要实现步骤描述如下:

a)参数初始化。初始化参数 N_{ed}(迁移次数)、N_{re}(繁殖次数)、N_{ch}(趋化次数)、P_{ed}(基本迁移概率)、S(细菌规模数,为了繁殖的需要,一般取偶数)、N_S(游动次数)、ω、C_1、C_2、α、β 等参数。

b)初始化细菌位置。采用式(4)产生初始化位置,初始化细菌的适应度函数值 J。

$$X=x_{\min}+\zeta\times(x_{\max}-x_{\min}) \tag{4}$$

式中,$\zeta\in[0,1]$是均匀分布的随机数。

c)迁移循环为 $E=1:N_{ed}$,繁殖循环为 $R=1:N_{re}$,趋化循环为 $C=1:N_{ch}$。细菌 i 的信息用 D 维向量表示为 $\theta^i=[\theta_1^i,\theta_2^i,\cdots,\theta_D^i]$,$i=1,2,\cdots,S$。$\theta^i(j,k,g)$ 表示细菌 i 在第 j 次趋化操作、第 k 次繁殖操作和第 g 次迁移操作之后的空间位置。

d)执行细菌趋化循环

(a)翻转操作。

①对所有细菌按照适应度函数值分成两类,优于适应度函数平均值的,通过追踪最优细菌方向进行局部细化搜索;反之,通过追踪细菌群体的中心位置进行全局搜索。

②以上两种方式都不满足时,按照式(5)和(6)更新细菌的位置。

$$\theta^i(j+1,k,g)=\theta^i(j,k,g)+C(i)\times\varphi(j) \tag{5}$$

$$\varphi(j)=\Delta(i)/\sqrt{\Delta^{\mathrm{T}}(i)\Delta(i)} \tag{6}$$

式中:$C(i)>0$ 表示向前游动的步长单位;$\Delta(i)\in[-1,1]$表示变向中产生的任意方向向量;$\varphi(j)$ 表示旋转后选取的一个随机前进方向的单位步长向量。

(b)游动操作。如果翻转的适应度函数值改善,则按照翻转的方向进行游动,直到适应度函数值不再改善或者达到预定的最大游动次数 N_s 为止。

$P(j,k,g)=\{\theta^i(j,k,g)\mid i=1,2,\cdots,S\}$ 表示细菌种群的位置,$J(i,j,k,g)$表示细菌 i 在第 j 次趋化操作、第 k 次繁殖操作和第 g 次迁移操作之后的适应度函数值,种群细菌之间传递信号的影响值为 J_{cc},如式(7)所示。

$$\begin{aligned}J_{cc}(\theta,P(j,k,g))&=\Sigma_{i=1}^{s}J_{cc}(\theta,\theta^i(j,k,g))=\\&\Sigma_{i=1}^{s}[-d_{\mathrm{att}}\times\exp(-\omega_{\mathrm{att}}\Sigma_{d=1}^{D}(\theta_d-\theta_d^i)^2)]+\\&\Sigma_{i=1}^{s}[h_{\mathrm{rep}}\times\exp((-\omega_{\mathrm{rep}}\Sigma_{d=1}^{D}(\theta_d-\theta_d^i)^2)]\end{aligned} \tag{7}$$

式中:d_{att}表示吸引力深度,ω_{att}表示吸引力宽度,h_{rep}表示排斥力高度;ω_{rep}表示排斥力宽度[5]。执行一次趋化操作后,细菌的适应度函数值按照式(8)进行更新。

$$J(i,j+1,k,g)=J(i,j,k,g)+J_{cc}(\theta^i(j+1,k,g),P(j+1,k,g)) \tag{8}$$

e)繁殖循环。趋化周期完成后,对每个细菌在生命周期内的适应度进行累加得到细菌能量,按照细菌能量进行排序,淘汰能量获取能力差的半数细菌,对能量获取能力强的半数细菌进行自我复制,各自生成一个与原有细菌完全相同的新细菌。新细菌与原有细菌具有相同的位置,即具有相同的觅食能力。

f)迁移循环。繁殖算子完成后,生成一个随机概率,并将它与基本迁移概率 P_{ed}进行比较,如果小于 P_{ed}则这个细菌个体死亡。在解空间内按照式(4)初始化,生成一个新的细菌个体,这个新个体与灭亡的个体可能具有不同的位置、不同的觅食能力。新的细菌个体可能更靠近全局最优解,有利于跳出局部最优解和寻找全局最优解。

g)循环结束条件判断,满足则结束,输出结果。

2　基于 PSO-BF 优化算法的关系数据库水印算法

2.1　研究前提

假设在数据使用范围内,对某些数据进行微小改动不会影响数据的使用。关系数据库的元组中某些数值型属性能够容忍由于嵌入水印而带来的微小变化,则称该属性为候选属性,候选属性的取值为实数。假设数据库元组中的主键是不能被修改的。设待嵌入水印的数据库关系为 DB($P,A_1,A_2,\cdots,A_N$),其中 P 表示主键,$A_j(1\leqslant j\leqslant N)$是第 j 个数值型属性,N 是 DB 中属性的个数。

2.2　水印信号的生成

为了提高水印的安全性,给定混沌初值(即密钥 K_1)生成混沌随机序列 L,截取 L 与二进制比特水印序列 I 等长(设水印序列的长度为 l),通过异或运算加密为待嵌入的水印信号 $W=L\oplus I$。

2.3　水印信号的嵌入

在嵌入水印之前,需要首先对属性值进行编号,依据编号的性质嵌入水印。采用单向 hash 函数将 DB 分成 m 个互不重叠的分组 $M=\{M_0,M_1,\cdots,M_{m-1}\}$,要求 $m>>l$,可以确保将水印信号多次重复嵌入到数据库中。选择候选属性中的一项属性名作为分组属性,采用单向 hash 函数将其映射为$\{0,1,\cdots,m-1\}$的值域。如果 hash 函数值为 i,则该元组放入分组 M_i 中。为了实现算法的鲁棒性,编号的计算还需要取决于密钥 K_2,即只有知道密钥 K_2 才能正确计算编号的大小。对于任意一个元组,其分组属性值为 $r.A_i$,分组属性名为 A_i,所在元组的

主键为 $r.P$,其编号 I_d 按照式(9)计算得到。

$$I_d = H(K_2, r.P, A_i) \bmod (m) \tag{9}$$

式中,$H(\cdot)$表示单向 hash 函数。

因此,分组 M_i 可以表示成多维分组向量 $M(i,p,d)$,其中 i 是分组编号,p 是第 i 个分组中将要嵌入水印的属性,d 是第 i 个分组中的元组。每个分组中嵌入 1 位水印信号比特 b_i,具体的嵌入策略用式(10)表示。

$$M^w(i,p,1:T_i) = M(i,p,1:T_i) + \delta(i,p,1:T_i) \tag{10}$$

式中:$M(i,p,1:T_i)$表示分组数据;$M^w(i,p,1:T_i)$表示嵌入 b_j 后的分组数据;$\delta(i,p,1:T_i)$表示对第 i 个分组中第 p 个属性的修改量,T_i 表示第 i 个分组第 p 个属性中数值型数据的总个数。$\delta(i,p,1:T_i)$受到数据使用范围 $U(i,p,1:T_i)$的约束,$U(i,p,1:T_i)$表示数据库元组中的候选属性在不影响使用的前提下能够容忍的变化范围。根据水印信号和数据库的统计特性来选择修改量 δ 的最优值,水印的嵌入过程抽象为约束条件下的优化问题。通过以下步骤来计算该优化问题的隐函数[6]:

a)通过式(11)计算第 i 个分组中第 p 个属性的参考点 R_i。

$$R_i = \mu_i + c \cdot \sigma_i \tag{11}$$

其中:$c \in [0,1]$是一个实数并且隶属于由数据库拥有者确定的秘密参数集合 γ;μ_i 和 σ_i 分别表示 $M^w(i,p,1:T_i)$的均值和标准差。

b)初始化,令 sum = 0。

c)对于 $M^w(i,p,j)$,$j = 1:T_i$。如果 $M^w(i,p,j) \geqslant R_i$,则 sum = sum + 1。

d)按照式(12)定义隐函数。

$$J_\gamma(M_i^w) = \text{sum}/T_i \tag{12}$$

使用 PSO-BF 优化算法来解决该优化问题,适应度函数选为式(12)定义的隐函数。细菌的初始化位置为随机放置在 T_i 维解空间 $\delta(i,p,1:T_i)$中,根据式(11)和(12)算出第 i 个分组中第 p 个属性的参考点和隐函数;类似地,再算出第 i 个分组中所有候选属性的隐函数,最终算出所有分组的隐函数。b_i 决定隐函数是最大化还是最小化,当 $b_i = 0$ 时,PSO-BF 优化算法使隐函数最小化;当 $b_i = 1$ 时,PSO-BF 优化算法使隐函数最大化。与 b_i 对应的隐函数最小值或最大值分别存入 j_{min} 和 j_{max} 中,这些值将在阈值解码中用于计算解码参数。在 $U(i,p,1:T_i)$约束条件下,使隐函数最优(最大或最小)的修改量 δ 为全局最优修改量 δ^*,从而不影响嵌入水印后的数据库数据的正常使用。

2.4 水印序列的提取

提取是嵌入的逆过程,通过已嵌入水印的数据库 DB_w 和秘密参数集合 γ 提取出水印序列。本文设计的是鲁棒性盲水印算法,即使 DB_w 被修改有失真(用 DB'_w 表示),不依赖于原始数据库,依然能够从 DB'_w 中提取出表征版权的水印序列。水印序列的提取可以划分为以下四个主要过程:

a)hash 函数分组。与嵌入过程一样,采用单向 hash 函数将 DB'_w 分成 m 个互不重叠的分组 M'。输入为:DB'_w,K_2 秘钥,分组属性和分组数目 m;输出为分组数据 $M' = \{M'_0, \cdots, M'_{m-1}\}$。

b)阈值解码。水印信号的比特解码基于使误码率最小的判决阈值 T^*[6]。误码率是指对已嵌入的水印比特进行错误解码的概率。计算 M'的隐函数 $J_\gamma(M'_i)$并与 T^* 比较,按照式(13)进行阈值解码。

$$\begin{cases} b'_i = 0 & \text{if } J_\gamma(M'_i) \leqslant T^* \\ b'_i = 1 & \text{if } J_\gamma(M'_i) > T^* \end{cases} \tag{13}$$

式中,b'_i 表示经过阈值解码获得的二进制水印信号比特。

设 P_e 表示误码率,P_0 表示 b_i 等于 0 的概率,P_1 表示 b_i 等于 1 的概率,b_e 表示嵌入的水印比特,b_d 表示解码的水印比特,$f(x)$表示概率密度函数,T 表示判决阈值。P_e 由式(14)计算得到。

$$\begin{aligned} P_e &= P(b_d = 0, b_e = 1) + P(b_d = 1, b_e = 0) = \\ &P(b_d = 0 \mid b_e = 1)P_1 + P(b_d = 1 \mid b_e = 0)P_0 = \\ &P(X < T \mid b_e = 1)P_1 + P(x > T \mid b_e = 0)P_0 = \\ &P_1 \int_{-\infty}^{T} f(x \mid b_e = 1)\,\mathrm{d}x + P_0 \int_T^{\infty} f(x \mid b_e = 0)\,\mathrm{d}x \end{aligned} \tag{14}$$

通过对 P_e 求驻点(即求一阶偏导数为零的点)获得 T^*,其中 P_e 对 T 的一阶偏导数 $\partial P_e/\partial T$ 如式(15)所示。

$$\begin{aligned} \partial P_e/\partial T &= P_1 \partial \int_{-\infty}^{T} f(x \mid b_e = 1)\,\mathrm{d}x/\partial T + P_0 \int_T^{\infty} f(x \mid b_e = 0)\,\mathrm{d}x/\partial T = \\ &P_1 f(T \mid b_e = 1) - P_0 f(T \mid b_e = 0) \end{aligned} \tag{15}$$

根据 j_{min} 和 j_{max} 的统计特性,得出 $f(x \mid b_e = 0) \sim N(\mu_0, \sigma_0)$,$f(x \mid b_e = 1) \sim N(\mu_1, \sigma_1)$和 $P_0 = 1 - P_1$,式(15)推导为式(16)的形式。

$$\begin{aligned} \partial P_e/\partial T = &\frac{P_1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(T-\mu_1)^2}{2\sigma_1^2}\right) - \\ &\frac{P_0}{\sigma_0 \sqrt{2\pi}} \exp\left(-\frac{(T-\mu_0)^2}{2\sigma_0^2}\right) \end{aligned} \tag{16}$$

令 $\partial P_e/\partial T = 0$ 可获得式(17)所示的二次方程式,该方程式的根 T_s 包含 T^*。

$$\begin{aligned} &\frac{\sigma_0^2 - \sigma_1^2}{2\sigma_0^2\sigma_1^2} T_S^2 + \frac{\mu_0\sigma_1^2 - \mu_1\sigma_0^2}{\sigma_0^2\sigma_1^2} T_s + \\ &\ln\left(\frac{P_1\sigma_1}{P_1\sigma_0}\right) + \frac{\mu_1^2\sigma_0^2 - \mu_0^2\sigma_1^2}{2\sigma_0^2\sigma_1^2} = 0 \end{aligned} \tag{17}$$

满足式(18)的 T_s 即为最优判决阈值 T^*。

$$\frac{\partial^2 P_e(T_s)}{\partial T^2} > 0 \tag{18}$$

c)择多表决法。由于在嵌入时,每一位水印比特 b_i 被重复嵌入到不同的数据分组中,相当于重复纠错码,可有效地降低误码率。采用文献[7]描述的择多表决法对 b_i 进行解码,得到密文水印 W',需要通过混沌映射解密为明文水印 I'。

d)混沌映射。与水印信号的生成过程一样,根据混沌初值(密钥 K_1)和水印序列的长度 l,生成混沌随机序列 L。通过异或运算将 W'解密为水印序列 $I' = L \oplus W'$。

3 实验仿真及结果分析

仿真实验所用到的数据库是关于某地区的地球化学测量数据,包含地理位置、样本特征、矿物、水质等参数的测量信息。该关系数据库由 77 212 个元组构成,每个元组有 287 个属性,实验环境为 MATLAB 7.10。

3.1 主要参数设置

选取二进制水印序列 I 的长度 $l = 128$,数据库分组数目 $m = 1100$,参数 $c = 0.75$。PSO-BF 优化算法中的参数已经根据实验调整为最佳值,$N_{ed} = 2$,$N_{re} = 2$,$N_{ch} = 40$,$P_{ed} = 0.25$,$S = 10$,$N_S = 4$。

3.2 鲁棒性分析

本文所提水印算法是在数据使用范围约束条件下修改关系数据库中数值型属性值,因此对于修改最不重要位的攻击或操作具有鲁棒性。为了抵抗属性重排序攻击,在水印的嵌入和提取过程中选取分组属性和密钥 K_2 为秘密参数,并采用单向 hash 函数对元组的次序进行加密处理。

模拟子集删除(删除随机选取的 τ 个元组)、子集修改(随

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.