基于神经网络混沌吸引子的公钥密码算法安全性分析及其实现

刘年生[1], 郭东辉[2*]
(1. 集美大学计算机工程学院, 福建 厦门 361021; 2. 厦门大学电子工程系, 福建 厦门 361005)

摘要: 介绍一种基于神经网络混沌吸引子的 Diffie-Hellman 公钥密码算法. 在过饱和贮存的 Hopfield 神经网络模型中混沌吸引子与初始状态之间存在一种单向函数关系, 如果改变该神经网络的联结权矩阵, 混沌吸引子及其所应的初始状态吸引域会随之发生改变. 因此, 我们可以其联结权矩阵为陷门, 利用可交换的随机变换矩阵来改变神经网络的联结权矩阵, 实现一种新的 Diffie-Hellman 公钥加密算法, 即将随机变换矩阵作为私钥, 而将变换后的神经网络联结突触矩阵作为公钥. 为了说明这种新公钥加密方案的实用性, 本文还分析和讨论其安全性和加密效率, 并利用 Java 编程实现互联网的应用方案. 实验结果表明, 本算法是可行的, 并具有较高的数据加密和解密速度.

关键词: 神经网络; 公钥密码体制; 混沌吸引子; 矩阵分解
中图分类号: TP 309.2 文献标识码: A 文章编号: 0438-0479(2007)02-0187-07

自 1976 年 Diffie 和 Hellman 首次提出公钥密码体制以来[1], Diffie-Hellman 公钥密码体制因其不需要安全信道来进行密钥的分配与传送, 并且在多用户保密通信时可有效地减少密钥数量, 方便和简化了密钥管理工作, 因此, 它倍受计算机网络安全通信的重视, 人们依此已经提出了多种公钥密码算法[2-3].

Hopfield 神经网络是一种简单结构的非线性系统, 但是它具有复杂的动力学特性和快速并行处理能力, 在密码学中具有很好的应用价值[4], 目前国内外的研究主要集中在其对称密码算法的设计上[5-8]. 我们曾经也利用过饱和 Hopfield 神经网络的混沌吸引子的特性[9], 提出了一种对称的几率加密算法[8]. 最近, 我们又根据 Diffie-Hellman 公钥密码体制, 以神经元触突联结矩阵作为陷门提出了一种新的基于神经网络混沌吸引子的公钥密码算法[10].

本文是在原有的工作基础上具体分析我们所提出这种新公钥算法的安全性, 并介绍采用 Java 编程语言在互联网中实现该算法的基本方案.

1 算法加密原理

1.1 神经网络模型

Hopfield 神经网络(Hopfield Neural Networks, HNN)是 Hopfield 在 20 世纪 80 年代初提出的一类神经网络模型[11], 可以进行硬件实现. 对于离散 Hopfield 神经网络而言, 如果神经网络的某一初始状态根据最小汉明距离(Minimum Hamming Distance, MHD)规则收敛到一个系统吸引子, 那么它就是稳定状态, 这些稳定状态通常被作为 HNN 的联想贮存样本. 但是联想神经网络的记忆容量是有限的, 对于由 N 个神经元组成的 HNN 而言, 对随机样本的记忆, 其存贮容量仅约为 0.14 N. 当所要存贮的样本数超过该模型的存贮容量, 那么该神经网络系统的稳定吸引子将发生畸变, 使得系统不能按汉明距离最小规则进行联想, 出现了过饱和存贮的混沌吸引性质, 这时的 HNN 就变成为 OHNN(Overstoraged HNN).

假设离散 Hopfield 神经网络有 N 个互联神经元, 每个神经元状态只为 0 或 1, 它的下一个状态 $S_i(t+1)$ 取决于当前各神经元的状态, 即:

$$S_i(t+1) = f\left(\sum_{j=0}^{N-1} T_{ij} S(t) + \theta_i\right), \quad i = 0, 1, 2, \cdots, N-1 \tag{1}$$

公式(1)中, T_{ij} 为神经元 i 与 j 之间的联接权值, θ_i 为神经元 i 的阈值, $f(x)$ 为任一非线性函数. 不妨设 $f(x) = \sigma(x)$ 为一符号函数, 则:

$$\sigma(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

在 HNN 模型中, 神经元的阈值 θ_i 可定义为 0(i =

收稿日期: 2006-06-06
基金项目: 国家自然科学基金(69886002, 60076015), 福建省自然科学基金(2006J0408), 福建省青年创新基金(2005J034), 福建省教育厅科技项目(JA05293)和集美大学优秀青年骨干教师基金(2006B003)资助
* 通讯作者: dhguo@xmu.edu.cn

0, 1, 2, …, $N-1$), T_{ij} 为一对称矩阵 T. 根据公式(1)和(2), 则有:

$$S(t+1) = F_T(S(t)) = \sigma(S(t)T) \tag{3}$$

其中 $S(t+1) = \{S_0(t), S_1(t), S_2(t), \cdots, S_{N-1}(t)\}$, 系统在时间 t 时刻时状态为:

$$S(t) = F_T(S(t-1)) = F_T^t(S(0)) \tag{4}$$

其在 t 时刻时系统的能量函数为:

$$E(t) = -\frac{1}{2}\sum_{ij} T_{ij} S_i(t) S_j(t) \tag{5}$$

Hopfield 已证明能量函数随系统状态的演进而单调下降[12], 由于神经网络的能量是有限的, 它最终会达到一种稳定状态, 即吸引子; 而郭东辉和郑利明等人的进一步研究结果表明它是混沌吸引子, 吸引子与初始状态之间不按 MHD 规则进行联想, 每个吸引子的吸引域中的状态消息之间存在不可预测的关系; 如果改变联结权值矩阵 T, 则吸引子及其相应的吸引域都会随之发生改变. 在引入随机变换矩阵 H 后, 原初始状态 S 和吸引子 S^{μ} 分别变为新的初始状态 $\hat{S}$ 和吸引子 $\hat{S}^{\mu}$[10]:

$$\hat{S}^{\mu} = S^{\mu}H \tag{6}$$

$$\hat{S} = SH \tag{7}$$

1.2 基于混沌吸引子的 Diffe-Hellman 公钥体制

根据矩阵理论[13], 当联结突触矩阵 T 为 n 阶奇异方阵时, 假设任取一 n 阶可对角化随机变换矩阵 H, 并保密, 则计算 $\hat{T} = HTH'$ 是容易的, 并且它是矩阵 T 的相合矩阵, 也是 n 阶奇异方阵. 同时, 在随机变换矩阵中存在一类特殊的矩阵族, 即可交换矩阵族, 假设 H_1 和 H_2 为可交换矩阵族中任意两个同阶方阵, 则它们满足 $H_1H_2 = H_2H_1$.

根据 Diffie-Hellman 公钥密码体制的思想, 在一组通信用户中共同选取一个联结突触矩阵 T_0, 它为 n 阶奇异方阵. 每个用户在 n 阶方阵交换族中随机选取一个变换方阵, 如用户 A 任意选取一个非奇异变换方阵 H_a, 首先计算 $T_a = H_aT_0H'_a$, 然后将 H_a 保密, 而把 T_a 公开. 当同一组内的用户 A 与 B 需要保密通信时, 他们就可以把 $\hat{T} = H_aT_bH'_a = H_bT_aH'_b$ 作为他们之间保密通信的共同密钥, 用户 A(或用户 B) 均可以根据自己的私钥和对方的公钥很容易地计算出公共密钥. 但是第三者将很难从公钥 T_a 和 T_b 中直接计算出 $\hat{T}$ 或 H_a 和 H_b, 特别当 n 较大时.

在 Diffie-Hellman 公钥密码系统中, 易于遭受中间人(Man int the middle) 攻击, 即第三者通过欺骗手段(如 ARP 欺骗、DNS 欺骗等等) 冒充合法的通信者. 因此, 在本公钥密码设计时, 为了进一步增强信息传输安全, 防止中间欺骗者攻击, 采用带认证的 Diffie-Hellman 密钥交换协议, 对保密通信的双方用数字签名和公钥证书来相互认证对方的身份是否合法[14].

2 加密方案

由上述神经网络过饱和存贮的混沌吸引性质可以知道: 只要改变少量的存贮样本(S^{μ}) 或少量神经元之间的连接矩阵元(T_{ij})[9], 该神经网络系统就可以获得具有大范围混沌的、随机的吸引域的分类吸引子. 为此, 我们可以根据 Diffie-Hellman 公钥密码体制设计出安全性较高的计算机公钥加密通信系统, 如图 1 所示, 其密钥产生与分配、加密过程和解密过程参见文献[10].

现以由 8 个神经元所组成的 OHNN 为例, 假设某用户组内所选的公共联结突触矩阵 T_0 为如下方阵[9]:

$$T_0 = \begin{bmatrix} 1 & -1 & 0 & 1 & -1 & -1 & 0 & 1 \\ -1 & 1 & 1 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ 1 & 0 & 1 & 1 & -1 & 0 & -1 & -1 \\ -1 & -1 & 0 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & -1 & -1 & 1 & 0 & 1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 1 & -1 & 0 \end{bmatrix}.$$

不妨设发送方和接收方的私有密钥分别为如下随机变换矩阵 H_s 和 H_r:

$$H_s = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$H_r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

发送方和接收方的公钥分别为:

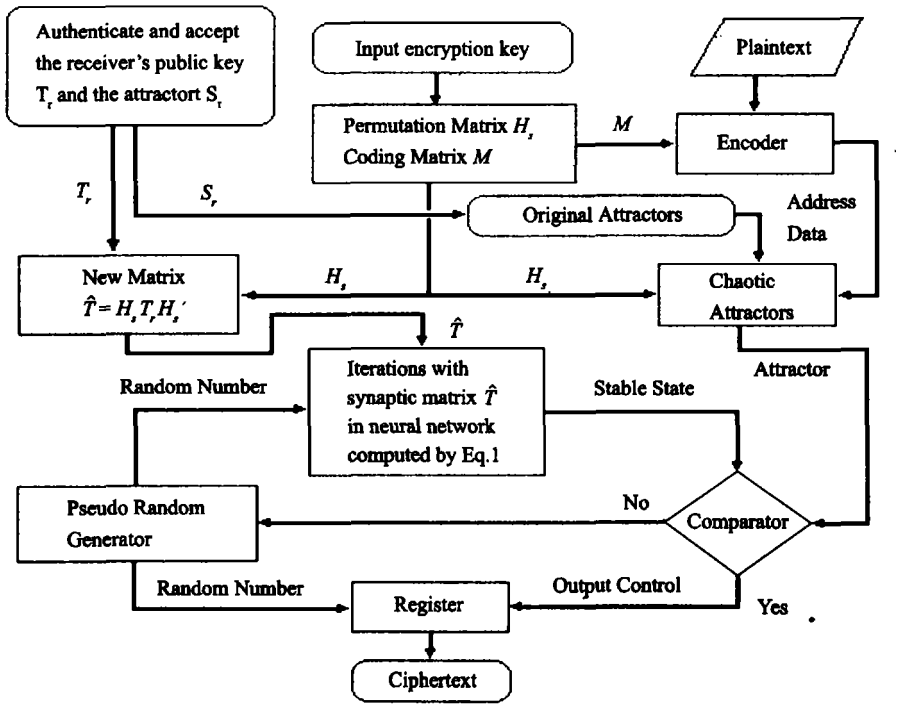

图 1　基于神经网络混沌吸引子的加密通信系统方案

(a) 加密算法;　(b) 解密算法

Fig. 1　*Scheme of encrypted communication based on the chaotic attractors of neural networks*

$$T_s = H_s T_0 H'_s =$$
$$\begin{bmatrix} 1 & 1 & 0 & -1 & -1 & 1 & 0 & -1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & 0 & 1 & 1 & 1 & -1 \\ -1 & -1 & 0 & 1 & 1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 1 & 1 & -1 & 0 & 1 \\ -1 & 0 & 1 & -1 & -1 & 0 & 1 & 1 \end{bmatrix},$$

$$T_r = H_r T_0 H'_r =$$
$$\begin{bmatrix} 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 & -1 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 & 1 & 1 & -1 & -1 \\ -1 & 0 & 1 & 1 & 0 & 1 & -1 & -1 \\ -1 & -1 & 1 & 0 & 1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & -1 & 0 & 1 & 1 & -1 \\ 1 & 0 & -1 & -1 & 1 & 0 & -1 & 1 \end{bmatrix}.$$

而通信双方的共同密钥(新的联结突触矩阵 $\hat{T}$)为:

$$\hat{T} = H_s H_r T_0 H'_r H'_s = H_r H_s T_0 H'_s H'_r =$$
$$\begin{bmatrix} 1 & -1 & 0 & -1 & -1 & 1 & 0 & 1 \\ -1 & 0 & 1 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & 1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 1 & -1 & 0 & 1 & -1 \\ -1 & 1 & 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 & 1 & 1 & -1 & 0 \\ 0 & -1 & -1 & 1 & 0 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & -1 & 0 & 1 & 1 \end{bmatrix}.$$

对应不同的联结突触矩阵如: T_0、T_s、T_r 和 $\hat{T}$ 等,均有相应不同的吸引子和吸引域,其中对于 $\hat{T}$ 而言,选择出吸引域较大的 8 个吸引子,则有 $\hat{S}^0$ = (11001011)、$\hat{S}^1$ = (11001101)、$\hat{S}^2$ = (01011101)、$\hat{S}^3$ = (01111100)、$\hat{S}^4$ = (01110110)、$\hat{S}^5$ = (10110110)、$\hat{S}^6$ = (10110011)、$\hat{S}^7$ = (10101011). 若采用 3 位的编码矩阵 M,即: 000 → 0, 001 → 1, 010 → 2, 011 → 3, 100 → 4, 101 → 5, 110 → 6, 111 → 7,则它们分别对应前面的 8 个吸引子 $\hat{S}^{\mu}$. 当加密端输入的明文为 110 时,它所对应的吸引子为 $\hat{S}^6$ = (10110011),再从 $\hat{S}^6$ 的吸引域中随机选择一个非稳定状态如(10010010)作为它对应的密文输出. 而在解密端,首先对发送方的身份进行认证,若合法,则输入自己的私有密钥和对方的合法公钥,计算出共同的密钥联结突触矩阵 $\hat{T}$,然后,当收到密文(10010010)时,根据公式(1)和联结突触矩阵 $\hat{T}$,计算出密文所对应的吸引子(10110011),即 $\hat{S}^6$ 为密文所相应的编码明文;再利用 3 位的编码矩阵 M, 6 → 110, 就恢复出原明文.

3 算法性能分析

评价一种加密算法优劣很重要的两个因素就是其所提供的安全性和加密速度,对本加密方案而言,数据加密速度主要受加密效率的影响.

3.1 加密方案的安全性

在我们所提出的公钥密码系统中,它的安全性是基于奇异矩阵的分解困难性和 OHNN 混沌分类特性. 对密码系统的攻击,其主要目的是寻找密钥,就本密码系统而言,它既可以根据 OHNN 混沌分类特性来攻击密钥,也可以通过对密码系统所采用的奇异矩阵进行分解来寻找密钥.

1) 对矩阵分解的攻击

加密方案中已说明了联结突触矩阵 T_0 为奇异方阵,因而 T_0、T_r、T_s 和 $\hat{T}$ 均为奇异方阵,由 T_0、H_r 和 H_s 计算 T_r、T_s 和 $\hat{T}$ 是容易的,而由 T_r、T_s 和 $\hat{T}$ 求解 H_r 和 H_s 是困难的,原因如下.

对于奇异方阵的分解而言,其分解不是惟一的,例如:假设奇异矩阵 $\hat{T} = B \times C$ 是 $\hat{T}$ 的一种分解,则对任何相同阶的左可逆矩阵 R(即 $R^{-1} \times R$ 等于单位矩阵), $\hat{T} = (B \times R^{-1}) \times (R \times C)$ 也是一种分解,其中 R^{-1} 为矩阵 R 的逆. 而且难以找到甚至不存在可行的求出全部分解的算法,因而从公钥 T_r、T_s 分别推出私有密钥 H_r 和 H_s 是困难的.

常见的矩阵分解方法主要正交三角分解(QR)分解、奇异值分解和三角分解(LU)分解. 第一, T_0、T_r、T_s 和 $\hat{T}$ 均为奇异方阵,因此它是不能通过 QR 分解方法来分解的. 第二,当 T_0、T_r、T_s 和 $\hat{T}$ 的阶数 n($n > 64$)较大时,利用矩阵的奇异值分解是行不通的,其困难性在于两个方面: a) 至今尚无切实可行的方法能求出一般高阶矩阵的全部准确特征值[15]; b) 奇异方阵的奇异值分解一般也不是惟一的,因为分解式中存在多个正交矩阵[16],至少有 $2 + 2^{2^{m-n+1}} + 2\sum_{k=2}^{r}(2^k!)^{2^{m-nk-1}}$,其中 m 为矩阵的阶,可令 $m > n$, $r = \frac{m-1}{n}$;而且,还应注意到正交矩阵具有如下两个重要性质: a) 正交矩阵的逆阵是正交矩阵; b) 任意两个正交矩阵的乘积仍是正交矩阵. 因此,尽管我们不知道 n 阶正交矩阵的确切个数,但仍可以初步推定其的空间范围是比较大的,遍历

其空间将是很困难的. 第三, 对于 LU 分解来说, 大多数情况下, 它们的分解并不是惟一的, 目前尚无法遍历所有的分解; 同时, LU 分解所得的分解并不是密钥产生那种的形如$\hat{T} = HT_0H'$ 的方式.

退一步讲, 即使第三者知道原突触联结矩阵 T_0, 采用试凑的方法, 要从公钥 T_i 中推出私钥 H_i 在计算上仍存在难以克服的困难性; 例如采用穷举攻击法来寻找私钥 H_i, 一种方式直接虚构一个 H_i, 测试 $H_iT_0H'_i$ 是否等于 T_i, 在这样情况下, 即使在 n 阶变换矩阵 H_i 中的所有元素只为 0 或 1, 那么, 它可能的数目为 2^{n^2}, 即它的计算时间复杂性为 $O(2^{n^2})$, 也即随矩阵阶数 n 的平方而呈指数性增长, 当 n 较大时, 由于计算量太大, 实际上是不可能计算的. 另一种方式就是采用矩阵变换, 这在变换矩阵 H_i 为正交矩阵时才能使用, 即先将 T_0 转化为 Hessenberg 矩阵, 然后将 T_i 也转化为 Hessenberg 矩阵, 而 T_i 与 T_0 可以具有一个相同的 Hessenberg 矩阵, 如果将 T_i 与 T_0 约化成同一 Hessenberg 矩阵, 则可以求出私钥 H_i 来, 但同样存在计算困难的问题, 其一是将任一方阵约化为 Hessenberg 矩阵, 其计算量为 $O(n^3)$, n 为矩阵的阶数; 其二是在一般情况下, Hessenberg 分解是不惟一的[13], 至少有 2^n 个. 因此, 当 n 比较大(如大于128)时, 要遍历其所有的 Hessenberg 分解形式在计算上是不可能的.

另外, 对于任何第三者他知道通信双方的公钥 T_s 和 T_r, 这样他是否能从中推出通信双方的公共密钥 $\hat{T}$. 对于这个问题, 第三者要知道公共密钥 $\hat{T}$ 只有两种途径, 一种就是从公钥 T_s(或 T_r)推出私钥 H_s(或 H_r)来求公共密钥 $\hat{T}$, 如前面所述将是很困难的, 当 n 比较大时, 计算上是不可行的; 另一种用已知的公钥 T_s 和 T_r 进行矩阵变换来公共密钥 $\hat{T}$, 如试求一矩阵 X, 让它满足:

$$\hat{T} = T_sXT_r \tag{8}$$

将其 T_s 和 T_r 代入公式(8), 则

$$\hat{T} = H_sT_0H'_sXH_rT_0H'_r \tag{9}$$

而

$$\hat{T} = H_sH_rT_0H'_rH'_s = H_rH_sT_0H'_sH'_r \tag{10}$$

由于 T_0 是奇异方阵, T、T_s 和 T_r 也都为奇异方阵, 不存在相应的逆矩阵, 所以不可能从理论上求解出一个矩阵 X 使得方程(9)与(10)相等, 这种想法也是行不通的.

2) 抗常用密码攻击的能力

目前无论是选择性明文攻击还是已知明文攻击都不可能找到其随机变换矩阵 H, 即密钥, 而且整个密码系统是不规则的, 在加密过程中它是随机选取密文的, 同一个明文块可对应多个密文块, 而在解密过程中又采用自吸引的方法. 差分密码分析法不可能有效地破译这种不规则的密码算法. 基于明文特性统计几率的穷举攻击法可能是惟一能破解本密码系统的有效方法, 但是代价是巨大的.

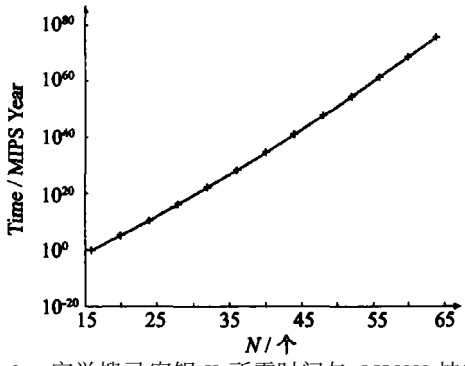

图 2　穷举搜寻密钥 H 所需时间与 OHNN 神经元个数之间的关系

Fig. 2　Time required to perform an exhaustive search of keys varies with the network scale

由前面的加密方案可知, 对由 N 个神经元所组成的 OHNN, 所选取的吸引子数目为 p, 则对于每种编码矩阵, 随机变换矩阵 H 有 $N!$ 种可能, 即密钥空间为 $N!$. 即使是已知明文攻击, 采用穷举法搜寻随机变换矩阵 H, 将要运行 $N!$ 次, 如果用计算能力为每秒 10^6 个变换矩阵 H 的专业计算机来穷举搜寻确定变换矩阵 H, 则遍历变换矩阵 H 空间所需的时间取决于网络的神经元个数 N, 如图 2 所示. 当 $N = 32$ 时, 成功地搜寻到一次变换矩阵 H 所需的就为 10^{20} MIPS Years 数量级, 高于目前可接受的安全水平 10^{12} MIPS Years.

另一方面, 在本密码方案的加密过程中, 要求在每个吸引域内随机选取一个非稳定状态 A^{μ} 来替代相应的吸引子, 以消除明文字符的统计相似性, 避免基于统计分析的密码攻击. 因而, 在加密过程中, 每个吸引域内非稳定状态的数目 Λ 对本密码系统的安全性而言是另一个密钥参数. 较大的 Λ 可降低相同明文出现相同密文的几率. 而参数 Λ 主要取决于 OHNN 的神经元个数 N 有关, N 越大, Λ 就越大, 例如, $N = 8$ 时, $\Lambda = 20$; 而 $N = 32$ 时, $\Lambda = 2^{16}$. 加密时替代吸引子的非稳定状态 A^{μ} 是用伪随机生成器产生的, 如果在我们所提出的密码系中将伪随机生成器设计成具有时变性, 则相同的明文在不同的时刻被加密成为不同的密文, 从而进一步增强了密码系统的安全性.

3.2　加密效率

我们所提出的这一加密方案与基于混沌同步的相比,它避免了同步混沌通讯系统中必须要求收发两端严格同步的诸多麻烦和不便,只要算法和密钥相同,就可以准确地进行信息的加密与解密. 同时,它采用 Diffie-Hellman 公钥密码体制,与对称密码密体制相比,可更好地满足现代计算机多媒体保密网络通信的需要,有效地减少用户之间的密钥量,方便了密钥管理. 不过它的密文长度比原明文的要长许多,存在着密文数据膨胀的问题. 从密码学的观点来说,一般不希望过度的密文数据膨胀,它会影响加密和传输的效率.

在本加密方案中,假设是由 N 个神经元所组成的 OHNN,每次加密 n bit 的二进制明文,则有 $p = 2^n$ 个 OHNN 的吸引子被作为替代 n bit 的二进制明文,(应注意的是:只有 OHNN 的吸引子样本数多于或等于 2^n 个,加密算法才能有效),每次所产生的密文的二进制长度为 N bit,则密文数据的膨胀率为:

$$e = \frac{N}{n} = \frac{N}{\log_2 p} \tag{11}$$

从公式(11)可以看出密文数据膨胀率与 OHNN 神经元个数和明文编码长度之间的关系,OHNN 神经元个数 N 越大,相应的吸引子数目就越多,如果采用适当的明文编码长度,就可有效的降低密文的膨胀率;又由于 OHNN 是采用并行运算模式,神经元个数的增多并不降低它的加密或解密的速度. 例如由 8 个神经元组成的 OHNN 可具有 17 个吸引子,选择其中 16 个吸引域较大的吸引子作为编码吸引子,就可采用 3 位的编码矩阵,密文数据膨胀率为 2.67;而由 16 个神经元组成的 OHNN 可具有 150 多个吸引子,若选择其中 128 个吸引域较大的吸引子作为编码吸引子,就可采用 7 位的编码矩阵,其密文数据膨胀率为 2.29,比前者降低了 0.38.

4 算法实现

利用 Java 编程语言对所提出的新公钥算法进行了软件实现. 在编程过程中,为了提高软件加密/解密速度,对算法中所出现的矩阵相乘,均采用 Strassen 矩阵乘法算法,这样,两个 n 阶矩阵相乘的计算复杂度由 $O(n^3)$ 降为 $O(n^{\log 7}) = O(n^{2.18})$.

软件实现的结果表明,本方案是可行的,常见的媒体类型包括文本、图片、视频和可执行程序等均可有效地进行加解密处理,实现互联网的保密通信,并进行本算法的数据加密速度测试,测试环境为:采用 DELL INSPIRON 6000 笔记本,CPU 为英特尔(R)奔腾(R) M 处理器 740 型,频率为 1.73 GHz,Java 编译器采用 JDK1.5 版,所测得的数据加密速率为(398.0±4.2) kB/s(p = 0.05),数据解密速度为(9 332.4±148.4) kB/s (p = 0.05),超过了 RSA 专用芯片的数据加密速度(5.7 kB/s)[17].

5 结 论

根据神经网络的混沌吸引子性质提出了一种新的公钥加密算法,从目前的安全性理论分析来看,该算法具有较高的安全性,可以有效地抵抗常规的密码分析方法的攻击,并利用神经网络在专用芯片中对敏感信息进行并行计算处理,数据加密速度比较高,其 Java 软件初步实现的数据加密速率比 RSA 芯片的还有要高. 因此,该加密算法为下一代互联网的安全通信提供一种新的候选加密算法,并且能以硬件方式实现并行处理,可以满足其对信息传输的安全性和实时性双重要求.

参考文献:

[1] Diffie W, Hellman M. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1976, IT-22(6): 644–654.

[2] William Stallings. Cryptography and network security: principles and practice[M]. 2nd. New Jersey: Prentice Hall Inc., 2003.

[3] Hellman M E. An overview of public key cryptography [J]. IEEE Communications Magazine, 2002, 40(5): 42–49.

[4] 郭东辉, 吕迎阳, 刘瑞堂, 等. 神经网络及其在网络通讯中的应用研究[J]. 厦门大学学报: 自然科学版, 2001, 40(2): 283–291.

[5] 齐锐, 张大力, 阎平凡. 基于神经网络的对称加密系统[J]. 清华大学学报: 自然科学版, 2001, 41(9): 89–93.

[6] Crounse K R, Yang T, Chua L O. Pseudo-random sequence generation using the CNN universal machine with applications to cryptography [C]// Proceedings of the IEEE International Workshop on Cellular Neural Networks and Their Applications. Piscataway: IEEE, 1996: 433–438.

[7] Veljko Milanovic, Mona E Zaqhloul. Synchronization of chaotic neural networks for secure communications [C]// IEEE International Symposium on Circuits and Systems. Piscataway: IEEE, 1996: 28–31.

[8] Donghui Guo, Cheng L M, Cheng L L. A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks[J]. Applied Intelligence, 1999, 10(1): 71–84.

[9] Gardner E. Maximum storage capacity in neural networks [J]. Europhys Lett, 1987, 4(4): 481–485.

[10] Liu Niansheng, Guo Donghui. A new public-key cryptography based on chaotic attractors of neural networks [C]// Progress in Intelligence Computation. Wuhan: China University of Geosciences, 2005: 293- 300.

[11] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities [J]. Proceedings of the National Academy of Science, USA, 1982, 79: 2554- 2558.

[12] Hopfield J J. Neurons, dynamics and computation [J]. Physics Today, 1994, 47: 40- 46.

[13] 陈景良, 陈向晖. 特殊矩阵[M]. 北京: 清华大学出版社, 2001.

[14] Bresson Emmanuel, Chevassut Olivier, Pointcheval David, et al. Provably authenticated group Diffie-Hellman key exchange[C]// Proceedings of the ACM Conference on Computer and Communications Security. Philadelphia: ACM Press, 2001: 255- 264.

[15] 程云鹏, 主编. 矩阵论[M]. 西安: 西北工业大学出版社, 2001.

[16] 温巧燕, 肖国镇. *m* 阶相关免疫函数的构造与计数[J]. 西安电子科技大学学报, 1997, 24(1): 36- 39.

[17] Alan Daly, William Marnane. Efficient architectures for implementing montgomery modular multiplication and RSA modular exponentiation on reconfigurable logic [C]// Tenth ACM International Symposium on Field-Programmable Gate Arrays. Philadelphia: ACM Press, 2002: 40-49.

Security Analysis of Public-key Cryptography Based on Chaotic Attractors of Neural Networks and Its Implementation

LIU Nian-sheng[1], GUO Dong-hui[2*]

(1. College of Computer Engineering, Jimei University, Xiamen 361021, China;
2. Department of Electronic Engineering, Xiamen University, Xiamen 361005, China)

Abstract: A new public-key cryptography based on chaotic attractors of neural networks is described in the paper. There is a one-way function between chaotic attractors and initial states in an Overstoraged Hopfield Neural Network (OHNN). If the neural synaptic matrix is changed with permutation operations, each attractor and its corresponding domain of attraction are simultaneously changed too. So we regard the neural synaptic matrix as a trap door and change it using commutative random permutation matrix. A new cryptography technique according to Diffie-Hellman public-key cryptosystem can be implemented. In the new scheme, the random permutation operation of the neural synaptic matrix is regarded as the secret key, while the neural synaptic matrix after permutation is regarded as public key. In order to explain the practicality of the proposed scheme, security and encryption efficiency of the new scheme are analyzed and discussed. The application scheme for Internet based on the proposed cryptography is implemented by using Java program. The experimental results show that the proposed cryptography is feasible and has a higher performance of encryption and decryption speed.

Key words: neural networks; public-key cryptosystem; chaotic attractor; matrix decomposition