基于无线传感器网络的混合混沌新分组加密算法*

佟晓筠† 左科 王蓁

(哈尔滨工业大学计算机科学与技术学院, 威海 264209)

(2011 年 5 月 6 日收到; 2011 年 6 月 9 日收到修改稿)

针对无线传感器网络 (WSNS) 中节点配备的能源少、节点计算能力低、存储资源有限以及传统的加密方法不适用于 WSNS 中等问题, 提出了一种新的基于动态迭代的混合混沌方程及其整型数值化方法, 并结合 Feistel 网络结构设计了一种快速、安全且资源消耗低的适用于 WSNS 节点的分组加密算法. 通过对混合混沌分组加密算法进行了大量的实验测试之后, 发现该算法具有密钥空间大、严格的雪崩效应、扩散及扰乱性高以及均等的统计平衡性等优点, 同时该算法还成功地通过了 SP800-22 的严格测试; 算法经过仿真器平台上运行的速度、时间及所占存储空间的测试分析, 结果表明设计的混合混沌分组加密算法是完全能够适用于 WSNS 节点的数据加密.

关键词: 无线传感器网络, 混合混沌, 分组算法, 数据加密

PACS: 05.45.Gg

1 引 言

无线传感器网络 (WSNS) 是由一组传感器以 Ad Hoc 方式构成的无线网络, 其目的是协作地感知、采集和处理网络覆盖的地理区域中感知对象的信息, 并发布给观察者 [1]. 传感器节点可以连续不断地进行数据的采集、事件检测和标识、位置监测以及节点的控制, 传感器节点的这些特性和无线通讯的连接方式使得无线传感器网络的应用前景非常广阔, 已被广泛应用于军事侦探、环境监测和预报、医疗卫生、智能家居、工业生产、复杂机械监控、城市交通、空间探索、大型车间和仓库管理, 以及机场、大型工业园区的安全监测等领域. WSNS 广泛的应用及其隐藏的巨大的潜在应用价值, 促使它成为下一代网络的研究和发展方向. 虽然各国已经开始了 WSNS 的研究, 但目前对 WSNS 的认识还处于初步发现和研究的阶段, 各种技术还不够成熟, 所以有必要对各种关键技术进行进一步的研究 [2,3].

由于传感器网络一般配置在恶劣环境、无人区域或敌方阵地中, 加之无线网络本身固有的脆弱性, 因而使得 WSNS 的安全问题比较突出, 所以要想在现实世界中大规模地应用 WSNS, 信息安全问题必须得到解决. 然而现阶段针对传感器网络的研究工作主要集中在硬件、无线网络技术和通信路由协议等方面, 而对数据安全方面的研究在国际上还处在起步阶段; 另一方面由于攻击者破密技术的发展, 过去认为安全的技术手段, 随着时间的推移已经难以保证安全, 并且与传统的 Ad-hoc 网络相比, WSNS 节点配备的能源少, 节点的计算能力低以及存储资源有限, 从而导致适用于 WSNS 节点的加密算法比较少; 而传统的加密方法如 DES (高级加密标准), AES (数据加密标准), RSA 等因其所需的硬件资源较多而不适用于 WSNS 节点的数据加密中, 同时由于现有的 WSNS 安全机制研究不足, 因而研究能够适用于无线传感器网络新的安全理论与加解密技术就显得很有必要了 [4]. 随着信息加解密技术的发展, WSNS 节点的加解密理论和技术就迫切需要得到更多的研究和发展; 同时随着混沌与传统密码学的相互作用、相互借鉴以及相互发展, 混沌逐渐被广泛地应用于加密领域, 这为混沌在 WSNS 节点的数据加密应用中提供了一个新的

* 国家自然科学基金 (批准号: 60973162)、山东省自然科学基金 (批准号: ZR2009RM037)、山东省科技攻关项目 (批准号: 2010GGX10132)、山东省重点自然科学基金 (批准号: Z2006G01)、哈尔滨工业大学 (威海) 校科学研究基金 (批准号: HITWHZB200909)、山东省威海市高新技术开发区科技发展计划 (批准号: 201025) 和山东省威海市科技发展计划 (批准号: 2008011) 资助的课题.

† E-mail: tong_xiaojun@163.com

发展方向 [5].

混沌现象是指在确定性的非线性系统中出现的一种貌似无规则、类似随机的现象, 是自然界普遍存在的复杂运动形式; 混沌不是简单的无序而是没有明显的周期和对称, 它是具有丰富的内部层次的有序结构 [6]. 混沌的轨道混合特性和混沌信号的类随机特性以及对系统参数的敏感性等特点, 使得混沌被普遍应用于密码学 [7−11] 与通信保密领域中 [12−15]. 随着混沌理论的不断发展, 混沌密码学的研究和应用领域也得到了不断的延伸; 同时现有的 WSNS 安全机制研究不足以及 WSNS 巨大的潜在应用价值, 使得将混沌应用于 WSNS 节点的数据加密成为可能. 在文献 [16] 中, 作者针对无线传感器节点资源有限等特点, 在基本的 Logistic 映射的基础上, 提出了一种适用于 WSNS 节点的整数混沌加密系统 (该加密系统简记为 LCS), 然而在文献 [17] 中作者针对该算法进行分析讨论后, 指出该算法很不完善, 安全性较低, 利用差分分析法就可将其破译.

本文针对以上情况, 根据 WSNS 中传感器节点的特点, 提出了一种基于 Devaney 混沌理论的混合整型数值化分组加密算法 (简称为 MCS), 该算法能够在传感器节点有限的资源下实现并满足加密数据的各种安全性以及时间、速度和存储空间等要求.

本文的主要内容安排如下: 第二部分, 主要研究了常应用于 WSNS 中的几种重要的传统加密算法, 其中包括 RC5，RC6，AES-Rijindael 以及 SKIPJACK 等算法的研究; 第三部分是适用于 WSNS 的混沌方程及其整型数值化研究; 第四部分是 WSNS 中基于动态迭代的混合混沌加密算法设计, 主要是基于动态迭代的混合混沌分组加密与解密算法的系统设计; 第五部分是实验验证和分析阶段, 该阶段主要是对实验数据进行安全性以及适用性等的分析; 第六部分总结本文所设计的加密系统.

2 传统加密算法研究

2.1 RC5 算法

RC5 是参数可变的分组密码算法, 可以表示为 RC5-$W/R/b$, 其中 W 表示待加密的字的比特长度, 通常可以取 16 bits, 32 bits, 64 bits; R 为整个加密过程要执行的轮数; b 表示加密密钥的字节长度, 一般取 128 bits, 但最大不能超过 255 bits. 它是 Ron Rivest 发明的, 由 RSA 实验室分析 [18,19]. RC5 算法是一种很简洁的算法, 它对数据的处理仅采用了非常适用于一般微处理器上运行的常见运算如模加、按位加、异或和循环移位等; 从其算法的设计可知, RC5 通过采用循环移位次数依赖于输入数据的技术, 实现了循环移位次数不能被预测的能力; 并且还具有所占存储空间少、运行速度快且轮数和密钥长度可变等特点. RC5 由于其所具有的特点满足了 WSNS 节点的加密需求, 从而被广泛应用于 WSNS 中. 理论研究分析表明 RC5 由于其扩散性不足, 存在一定安全隐患 [20]. 算法主要思想见文献 [21].

2.2 RC6 算法

RC6 是作为 AES(advanced encryption standard) 的候选算法提变给 NIST 的一种新的分组密码算法. 为了能更好地满足 AES 的安全要求, 它在 RC5 的基础上增加了对安全性的考虑. RC6 继承了 RC5 设计简单、广泛使用数据相关的循环移位思想; 同时为了增强其抵抗攻击的能力, RC6 改进了 RC5 中循环移位的位数不依赖于寄存器中所有位的不足 [22]. RC6 是对 RC5 进一步的发展, 加入了二次函数 $f(x) = x^*(2x + 1)$, 从而提高了密码扩散速度, 所以相对于 RC5 而言, RC6 可用较少的循环来增强其安全性. 同时由于 RC6 的处理块为 128 bits, 密钥长度可变, 是一个参数化的算法, 可以表示为 RC6-$W/R/b$, 其中 W 表示一个字的比特长度, 通常是 32 bits, R 代表加密算法所要执行的加密轮数, b 表示为密钥长度 (以字节为单位). 所以用户可以根据需要灵活地设定 W, R, b 以满足未来的发展和市场的需求. 文献 [23] 提出了该算法的主要设计思想.

2.3 AES-Rijndael 算法

AES 是美国国家标准与技术研究院 (NIST) 为了寻找一种能够用来替换 DES 而由 Joan Daeman 和 Vincent Rijmen 两位比利时科学家研究设计开发的算法. 由于该算法采用了 Rijndael 算法, 故又称 AES-Rijndael 算法. AES 算法采用的是 SP 结构代替置换网络结构; 分组的长度为 128 bits, 有三种可选的密钥长度分别为 128 bits, 192 bits, 256 bits, 轮数 r 依赖于密钥长度. 若密钥长度为 128 bits, 则 $r = 10$; 若密钥长度为 192 bits, 则 $r = 12$; 若密钥长度为 256 bits, 则 $r = 14$[24]. 每一轮由线性混合

层、非线性层和密钥加层组成, 线性混合层起扩散作用, 非线性层由 S 盒构成起混淆的作用, 密钥加层是简单的子密钥参与的异或运算, 加解密过程分别需要 r+1 个子密钥 [25]. 文献 [26] 提出了该算法的主要设计思想.

2.4 Skipjack 算法

Skipjack 是美国政府在 Clipper 芯片和 Fortezza PC 卡中使用的密钥加密算法. 它是一个密钥长度为 80 bits、明文和密文长度均为 64 bits、轮数为 32 的分组密码算法, 解密时密钥的输入与加密的输入顺序是一样的.

Skipjack 算法的加密过程包括 A, B 规则, 加密算法先将 64 bits 的明文分组块分成 4 个 16 bits 的明文子块 $w_i^0, 0 \leqslant i \leqslant 4$, 执行 A 规则 8 轮, 然后转入执行 B 规则 8 轮, 再执行 A 规则 8 轮, 再执行 B 规则 8 轮, 得到密文: $w_i^{32}, 0 \leqslant i \leqslant 4$; 解密时将输入的 64 bits 的密文分组块分成 4 个 16 bits 的密文子块 $w_i^{32}, 0 \leqslant i \leqslant 4$, 先执行 B^{-1} 规则运算 8 轮, 再执行 A^{-1} 规则 8 轮, 然后再执行 B^{-1} 规则运算 8 轮, 再执行 A^{-1} 规则 8 轮, 得到明文 $w_i^{32}, 0 \leqslant i \leqslant 4$, 文献 [27] 提出了该算法详细的设计思想.

3 适用于 WSNS 的混沌方程及其整型数值化研究

混沌学的发展离不开对具体混沌系统的研究, 同样, 混沌的应用研究也离不开具体的混沌系统, 其中帐篷映射和 Arnold 映射 (猫映射) 就是两种典型的实数域内的混沌系统. 然而由于 WSNS 节点的嵌入系统不擅长处理浮点、多字节除法等运算, 因而研究时域和幅度都离散化的混沌方程, 将有助于混沌加密在 WSNS 中的应用. 下面具体介绍帐篷映射和 Arnold 映射 (猫映射) 及其整型数值化方法.

3.1 帐篷映射及其整型数值化研究

帐篷映射的一般定义为

$$x_{n+1} = \begin{cases} \dfrac{x_n}{a}, & 0 \leqslant x_n < 0.5, \\ \dfrac{(1-x_n)}{(1-a)}, & 0.5 \leqslant x_n < 1. \end{cases} \tag{1}$$

该映射是通过引入可变参数 a 得到所谓的斜帐篷映射, a 的取值决定了帐篷顶点的位置, 当 $a = 0.5$ 时顶点在中间, 此时就变成了标准的帐篷映射. 通过进一步推广, 可以得到一类分段线性映射, 其方程为

$$x_{n+1} = \begin{cases} \dfrac{x_n}{p}, & 0 \leqslant x_n < p, \\ \dfrac{(x_n - p)}{(0.5-p)}, & p \leqslant x_n < 0.5, \\ \dfrac{(1-x_n-p)}{(0.5-p)}, & 0.5 \leqslant x_n < 1-p, \\ \dfrac{(1-x_n)}{p}, & 1-p \leqslant x_n \leqslant 1, \end{cases} \tag{2}$$

其中: x_n 表示第 n 次迭代的值, x_{n+1} 表示第 $n+1$ 次迭代的值, a 与 p 为参数.

帐篷映射的优异特性之一是其具有均匀的分布函数, 在实数域内其性态是混沌的, 具有伸长和折叠特性. 其伸长特性最终导致相邻点的指数分离, 即敏感的初始条件; 其折叠特性则使其保持生成序列的有界性, 且使映射不可逆. 如果将其由实数域上进行的运算等价地转化为整数域上的整数运算, 使其成为整数域上的帐篷映射, 则同样也会保持帐篷映射所具有的伸长和折叠特性.

对于形如 (2) 式的分段线性映射, 对 (2) 式两边同时乘以 a, 令 $p = 1/4$, $Z_n = ax_n$, 则有 $Z_{n+1} = ax_{n+1}$, 将其代入 (2) 式化简得 (3) 式

$$Z_{n+1} = \begin{cases} 4Z_n, & 0 \leqslant Z_n < \dfrac{1}{4}a, \\ 4Z_n - a, & \dfrac{1}{4}a \leqslant Z_n < \dfrac{1}{2}a, \\ 3a - 4Z_n, & \dfrac{1}{2}a \leqslant Z_n < \dfrac{3}{4}a, \\ 4a - 4Z_n, & \dfrac{3}{4}a \leqslant Z_n \leqslant a. \end{cases} \tag{3}$$

令 (3) 式中的定义域为整数域, 如果取 $a = 2^w$, w 为计算机字长, 则 Z_n 的取值范围正好是计算机字长能表示的无符号整数范围 (除去 2^w 断点), 那么 (3) 式就是在计算机字长表示的无符号整数范围内的整型迭代运算. 这样对于 (3) 式的整数计算只需要进行移位、乘法、加法 (求补) 即可完成. 因此, 采用 (3) 式的混沌整型迭代运算是非常适合于 WSNS 节点的嵌入系统的处理.

对于 (3) 式得到的全部整数迭代解而言, 可以得到两个稳定的零解, 分别为 0 和 a; 然而在有限的二进制位离散数字计算中, 由于量化误差的存在, 即使初始值不为零, 则由 (3) 式经过多次的迭代计算, 只要得到 a 值, 则以后再迭代的值就保持为零值. 为了消除这种现象, 达到改进整型分段线性映

射的性态, 必须对 (3) 式调整为

$$Z_{n+1}=\begin{cases}4Z_n+1, & 0\leqslant Z_n<\dfrac{1}{4}a且Z_n为奇数,\\ 4Z_n-1, & 0\leqslant Z_n<\dfrac{1}{4}a且Z_n为偶数,\\ 4Z_n-a, & \dfrac{1}{4}a\leqslant Z_n<\dfrac{1}{2}a,\\ 3a-4Z_n-1, & \dfrac{1}{2}a\leqslant Z_n<\dfrac{3}{4}a,\\ 4a-4Z_n+1, & \dfrac{3}{4}a\leqslant Z_n<a且Z_n为奇数,\\ 4a-4Z_n-1, & \dfrac{3}{4}a\leqslant Z_n<a且Z_n为偶数.\end{cases}\tag{4}$$

从 (4) 式中可以看出, $4a-4Z_n+1$ 不会出现大于等于 a 的情况 ($a=2^w$, w 一般是 8 的整数倍), 因为 Z_n 为奇数, 所以 $4a-4Z_n+1$ 整型迭代解的范围为 $[5,a-3]$. 同理可知, 只要在 (4) 式规定的范围内进行整型迭代操作, 那么其迭代解的范围为 $Z_n\in[0,a-1]$, 且不会出现稳定的零解. 在 (4) 式中通过控制 Z_n 的奇偶性来消除取值为 -1 和大于等于 a 的特殊情况. 同时由于 (4) 式是分段的线性映射, 所以在 $Z_n\in[0,a-1]$ 范围内肯定是满足封闭性定理.

所以基于以上分析讨论可知, 基于分段线性映射改进的整型混沌计算 (4) 式对于在整数范围 $[0,a-1]$ 内构成了一个整数代数系统.

3.2 Arnold 映射 (猫映射) 及其整型数值化研究

Arnold 映射的方程定义为

$$\begin{cases}x_{n+1}=x_n+y_n, \pmod 1\\ y_{n+1}=x_n+2y_n, \pmod 1\end{cases}\tag{5}$$

为了方便应用, 更习惯于把它写成矩阵形式为

$$\begin{pmatrix}x_{n+1}\\ y_{n+1}\end{pmatrix}=\begin{pmatrix}1&1\\ 2&1\end{pmatrix}\begin{pmatrix}x_n\\ y_n\end{pmatrix}\bmod 1,\tag{6}$$

其中: x_n, y_n 表示第 n 次的迭代值, x_{n+1}, y_{n+1} 表示第 $n+1$ 次的迭代值, 其变换系数矩阵为 $\boldsymbol{C}=\begin{pmatrix}1&1\\ 1&2\end{pmatrix}$, 并且必须满足其系数矩阵行列式等于 1, 因此猫映射是一个二维保面积、可逆映射、没有吸引子的映射. 猫映射包括拉伸和折叠两个过程, 乘以矩阵 $\boldsymbol{C}$, 使 x, y 变大, 相当于拉伸; 取模使 x, y 又折回单位矩阵内, 相当于折叠. 猫映射是一个混沌映射, 同时也是一个一一映射, 单位矩阵内的每一点通过变换将惟一地变换到矩阵内的另一点. 猫映射通常用于图像像素点位置的置乱, 但是通过观测后发现, 也可以很方便地通过将猫映射整型数值化而用于混沌整型加密系统中, 整型数值化后的猫映射依然保持着原先良好的拉伸和折叠特性. 令 (5) 或 (6) 式中的变量的定义域为整数域, 再对其进行求模即可, 整型数值化后的方程为

$$\begin{pmatrix}x_{n+1}\\ y_{n+1}\end{pmatrix}=\begin{pmatrix}1&1\\ 1&2\end{pmatrix}\begin{pmatrix}x_n\\ y_n\end{pmatrix}\bmod 2^w\tag{7}$$

其中 w 为计算机字长, 由于猫映射的拉伸和折叠使得 x_k, $y_k\in[0,2^w-1]$, 这表示猫映射迭代值的范围正好落在计算机字长所能表示的无符号整数范围内 (除去 2^w 断点), 所以 (7) 式就是在计算机字长所能表示的无符号整数范围内 (除去 2^w 断点) 的迭代运算, 这种运算也是适合于 WSNS 节点嵌入系统的处理.

对于 (7) 式得到的全部整数迭代解而言, 只有当 x_k, y_k 同时为零时才能得到一个稳定的零解. 为了消除出现的稳定零解, 达到改进整型猫映射的性态, 理论上只要让迭代输入值 x_k, y_k 不同时取零值即可, 所以在实际应用中可以通过检测下一次的输入值是否同时为零, 如为零, 则可以将其输入值修正为其他的值即可. 同时由于整型数值化猫映射的拉伸和折叠特性使得 (7) 式在 $[0,2^w-1]$ 范围内肯定是满足封闭性定理.

所以基于以上分析可知, 基于猫映射改进的整型混沌计算 (7) 式在整数范围 $[0,2^w-1]$ 内构成了一个整数代数系统.

4 WSNS 中基于动态迭代的混合混沌加密算法设计

我们知道, 在设计加密体系时有几类网络结构被经常使用, 它们分别是 Feistel 网络 (如 DES, FEAL, TWOFISH, LOKl97, GOST), 变型 Feistel 网络 (如 RC5, MISTY2, CAST-256) 以及 SP 网络 (如 IDEA, Rijndael, SAFER)[28]. 本文则提出了一种基于混合混沌映射 (分段线性映射与猫映射) 并采用 Feistel 网络结构的新的分组加密算法. 该算法使用了 CBC 的加解密模式 (初始向量也为 32 bits), 分组长度为 32 bits, 密钥长度为 128 bits, 加密轮数为 13. 由于一般的 WSNS 节点 CPU 的机器字长为 8 bits, 为了使算法能够更好更快地被节点的硬件所执行, 故要求所设计的加密算法能以 8 bits 为单元来对数据进行处理, 并且在进行计算的过程

中只能用到移位、乘法 (多数乘法是通过移位进行)、减法 (求补) 和加法等非常适合 WSNS 节点嵌入系统处理的整型运算操作.

4.1 混合混沌加密算法

简要地说, 本文所采用的 Feistel 网络结构加密算法就是在一个轮函数 F 的多次作用之下, 将相应的明文分组转换成对应的密文分组. 设一个 R 轮的 Feistel 网络结构加密算法, 其分组长度为 $2n$ bits, 那么每轮的操作可形式地定义如下: Round_i : $L_{i-1} \parallel R_{i-1} \mapsto R_{i-1} \parallel F(K_i, R_{i-1}) \oplus L_{i-1}$, 其中 $i = 1, 2, \cdots, r$, L_i 和 R_i 分别为分组的左右两个部分 (两部分的长度均为 n bits), K_i 为第 i 轮使用的加密子密钥, 由主密钥 K 通过某种算法产生; 其中 F 函数是整个加密体系的核心部件, 在 Feistel 网络结构中, 这个函数可以不必要求其是可逆的, 但通常必须是能够起到混淆作用的非线性函数.

本算法是在函数 F 中加入了基于动态迭代的混合混沌机制, 由于在该算法函数 F 的设计中实现了混沌方程动态选择迭代次数的技术, 以及混沌具有的良好密码学特性, 从而导致函数 F 的非线性行为将更加复杂和难以预测. 图 1 给出了本文设计的基于动态迭代的混合混沌加密算法的整体描述, 该算法一次能处理 32 bits 数据, 而且对这 32 bits 数据的处理都是以 8 bits 为单位来进行存储和操作的, 这样就可以直接调用节点 CPU 内置的基本的单字节 (8 bits) 运算指令来完成运算和存储等操作, 并且加密算法在执行操作的过程中只用到了移位、乘法 (多数乘法是通过移位进行)、减法 (求补) 和加法等非常适合 WSNS 节点嵌入系统处理的整型运算操作, 同时这些都是节点 CPU 内置的基本运算指令, 可以直接调用, 这就使得加密算法具有了耗能少、安全性高、耗时少以及运行快等优点. 算法能支持 128 bits 长度的密钥, 同时在时间和空间复杂度允许的情况下只需改动算法的一个控制参数就可实现动态地扩展密钥的长度而使得算法满足新的安全性要求.

整个加密算法共执行了 13 轮 (通过实验验证发现选择执行 13 轮是基于以下两点: 一是传感器节点的存储空间以及加密执行速度或时间; 二是考虑到了算法的安全性), 其中每轮都使用了一个加密子密钥 K_i, 而每个加密子密钥同时又包含了 4 个子密钥分量分别为 $K_i(1), K_i(2), K_i(3), K_i(4)$, 这些子密钥 $K_i(i = 1, 2, \cdots, 13)$ 都是主密钥通过一定的密钥扩展算法产生出来的. 同时为了增强算法的安全性, 每轮的 F 函数中的两个整型数值化混沌方程执行的轮数也是根据它们的初始输入值而动态地选择. 除第一轮和最后一轮没有交换外, 其余各轮都做相同的函数 F 运算. 与此同时在算法的开始与结尾处各增加了一个 32 bits 置换操作, 以此达到增强算法的扩散性.

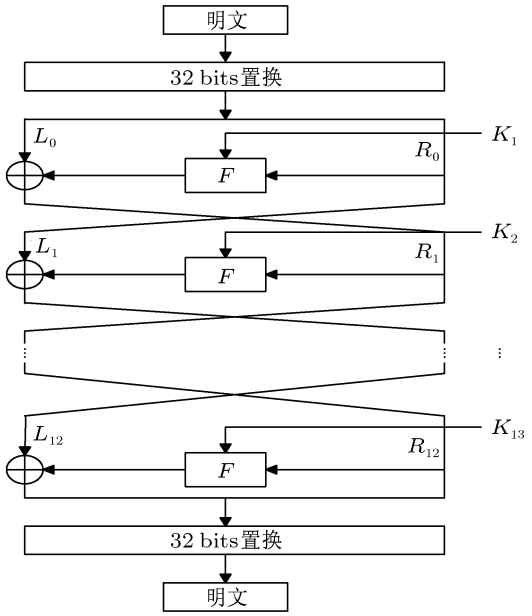

图 1 基于 Feistel 网络的混沌加密模型

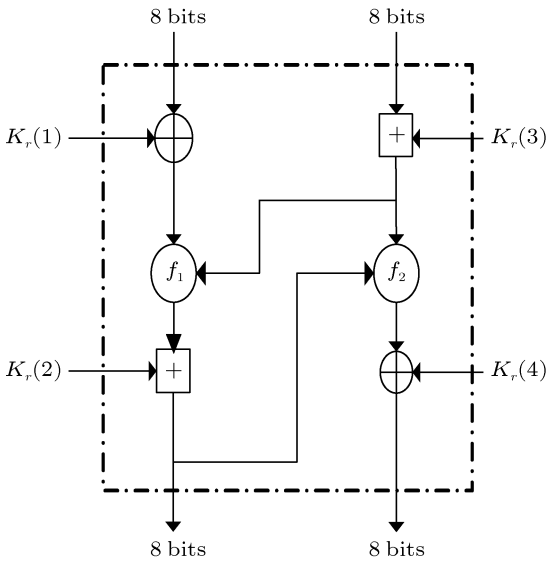

图 2 基于动态迭代的混合混沌的 F 函数

4.2 动态混合混沌的 F 结构

函数 F 的内部结构如图 2 所示, 其中 $\oplus$ 表示按位异或, $\boxplus$ 表示模 2^8 加法运算, 函数 f_1, f_2 分别表示整型数值化的猫映射方程 (7) 式和分

段映射方程 (4) 式, $K_r(t),(t=1,2,3,4)$ 代表第 r 轮加密子密钥的 4 个加密子密钥分量, 该函数 F 的实现主要是依赖于 f_1,f_2 函数也即方程 (7) 式和 (4) 式, 其中 f_1,f_2 两个整型数值化混沌函数执行的迭代轮数动态地依赖于它们的初始输入值, 也即在算法执行的每一轮中, f_1,f_2 这两个整型数值化混沌函数的迭代轮数都是根据它们的初始输入值来动态选择的, 目的就是增加扰乱和扩散效果. 函数 F 将 2 个 8 bits 的输入子分组在加密子密钥 $K_r(t),(t=1,2,3,4)$ 的控制下经过异或, 模 2^8 加法及方程 (7) 式和 (4) 式的作用而转换为 2 个 8 bits 的输出子分组, 其中第 i 轮加密子密钥 K_i 由主密钥 $K=K_1K_1\cdots K_{16}$ 按照 4.3. 节所示的密钥扩展算法生成.

4.3 密钥 $K_r(t)$ 扩展算法

密钥是密码算法设计中要考虑的一个重要因数, 在考虑密钥设计时应该尽可能地使加密子密钥与主密钥之间的关系复杂化, 目的就是增加由加密子密钥推出主密钥的难度, 保证加密体系以及密钥的安全性.

Ron Rivest 在 1994 年设计的 RC5 分组加密算法中采用的密钥扩展算法是通过采用初始主密钥作为密钥扩展参数的方法来控制加密子密钥的产生, 这种密钥扩展算法的优点就在于它能充分地将初始主密钥每一比特位的影响扩展到加密子密钥中去, 这就使得攻击者即使在获得子密钥的情况下也无法分析出初始主密钥, 从而提高了 RC5 密钥扩展算法的安全性能, 并且整个扩展密钥算法的运算量不复杂且容易在传感器节点上实现. 基于以上根据, 本文的密钥扩展算法将在 RC5 密钥扩展算法的基础上进行相关修改以满足 WSNS 节点运算的要求. 具体方案如下.

1) 初始化密钥向量 S

首先通过 RC5 的 P_w, Q_w 与 w 的关系, 以及固有的 P_w, Q_w 计算式得到当 $w=8$ 时的 P_8, Q_8 值 [29,30], 然后再通过以下伪代码初始化主密钥向量 S (其中 $t=16$, 为加密所需初始主密钥的字节数):

$$
\begin{aligned}
&S[0]=P_8;\\
&\text{for}(i=1;i<t-1;i++)\\
&S[i]=S[i-1]+Q_8.
\end{aligned}
$$

由于每轮的操作都要使用一个加密子密钥, 而每个加密子密钥中又包含了 4 个子密钥分量, 那么 13 轮迭代就需要 13 个加密子密钥, 而每个加密子密钥含 32 位, 理论上可以认为 S 就是一个 13×32 bits 的矩阵 (向量), 所以 S 向量就要占用 52 bytes 的存储空间. 虽然密钥长度越长安全性就越高, 所需的存储空间也就越大, 所以在考虑到 WSNS 节点空间有限以及算法执行所需的安全性、能耗、速度或时间等问题后, 本文采取了一种合理并且有效的折中方法, 即通过 $S[i]\rightarrow S[i\%16]$, $(i=0,1,\cdots,51)$ 对 S 向量下标进行这样简单的处理后既满足安全性又节省空间.

2) 生成 K_r 加密子密钥

将含有 16 个字节的初始主密钥 $K[0\cdots 15]$ 复制到含有 16 个字节的向量 $L[0\cdots 15]$, 然后再通过以下伪代码得到加密密钥向量 S (其中 $t=16$):

$$
\begin{aligned}
&A=B=0;\ \ i=j=0;\ \ k=3*t;\\
&\text{for}(;k>0;k--)\{\\
&A=\text{ROTL}(S[i]+A+B,3);S[i]=A;\\
&B=\text{ROTL}(L[j]+A+B,A+B);L[j]=B;\\
&i=(i+1)\%t;j=(j+1)\%t;\}
\end{aligned}
$$

其中 ROTL(X,n) 代表左循环移位: 将 X 循环左移 n 位.

4.4 解密算法

Feistel 网络结构密码的一个非常好的特征是具有相同的加密和解密结构, 这也意味着解密就是加密的逆运算而已, 所以解密只需使用与加密相反顺序的加密子密钥即可. 并且函数 F 可以设计得非常复杂而可以不要求它的可逆性, 因为 Feistel 网络结构的特点保证了分组加密算法无论函数 F 是否可逆都可以保证算法的可逆性, 即 $R_i\oplus F(L_i,K_i)=(L_{i-1}\oplus F(R_{i-1},K_i))\oplus F(L_i=R_{i-1},K_i)=L_{i-1}$.

5 实验验证及各种性能分析

5.1 扩散和混乱特性分析

Shannon[31] 在其经典文章中提出了设计分组密码体系的两条基本指导原则, 即扩散 (diffusion) 和混乱 (confusion) 原则. 扩散是将每一位明文的影响尽可能地作用到较多的密文位中去, 同时, 还要尽量使每一位密钥的影响也尽可能快地扩散到较多的密文位中去, 希望密文中的任一比特都要尽可能与明文、密钥相关联, 以便有效地隐藏明文的统计特性. 混乱则是用于掩盖密钥或明文与密文之间

的关系, 使密钥或明文与密文之间的统计关系尽可能地复杂化, 以避免很有规律的线性关系的出现.

在分组密码算法体制中, 算法的混乱扩散程度可以通过非线性扩散特性的统计检测给出, 加密算法的非线性扩散程度分析通常包括算法的完全性、雪崩效应、严格雪崩准则等三个方面. 完全性和雪崩效应首先由 Kam Davida 和 Feistel 分别介绍, 而 Webster 和 Tavares 则进一步提出了严格雪崩准则的概念.

所谓完全性是指加密函数输出的每一比特都与输入的所有比特 (包括明文与密钥) 有关; 所谓雪崩效应是指输入任一比特 (包括明文与密钥) 的改变都应造成输出平均半数比特的改变; 所谓严格雪崩准则是指输入任一比特 (包括明文与密钥) 的改变都应造成输出每一比特以 1/2 的概率发生改变.

如果设 F 是一个 n 比特输入, m 比特输出的变换函数, 记输入向量为 $X=(x_1,x_2,\cdots,x_n)$ 其中 $x_k\in\{0,1\},k=1,2,\cdots$. 仅改变 x 的第 i 比特后的输入向量为 $X^{(i)},i=1,2,\cdots,n$. 则它们对应的输出向量可分别记为二进制向量 $F(X),F(X^{(i)})$.

设函数 F 的输入变量取自样本子集 $T\subset Z_2^n$, 其中 Z_2^n 表示所有的 n 比特的二进制集合, 记 $\#T$ 代表输入向量个数. 记 $a_{ij}=\#\{X\in T|(F(X))_j\neq(F(X^{(i)}))_j\}$(其中 $i=1,2,\cdots,n;j=1,2,\cdots,m$) 表示 T 中的输入向量 X 和 $X^{(i)}$ 对应的输出向量之间第 j 比特不同的个数; $b_{ij}=\#\{X\in T|W_H(F(X)\oplus F(X^{(i)}))=j\}$(其中 $i=1,2,\cdots,n;j=0,1,...,m$) 表示 T 中的输入向量 X 和 $X^{(i)}$ 对应的输出向量之间的差分汉明重量为 j 的个数. 通过以上说明我们可以分别用 d_1,d_2,d_3,d_4 来表示算法非线性扩散程度的度量 [32]

$$d_1=\frac{1}{\#T*nm}\times\sum_{i=1}^{n}\Big(\sum_{X\in T}W_H(F(X)\oplus F(X^{(i)}))\Big),\quad(8)$$

$$d_2=1-\frac{1}{nm}\#\{(i,j)|a_{ij}=0\},\begin{pmatrix}i=1,\cdots,n\\j=1,\cdots,m\end{pmatrix},\quad(9)$$

$$d_3=1-\frac{2}{\#T*nm}\sum_{i=1}^{n}\left|\sum_{j=1}^{m}jb_{ij}-\frac{m}{2}\#T\right|,\quad(10)$$

$$d_4=1-\frac{2}{\#T*nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left|a_{ij}-\frac{1}{2}\#T\right|,\quad(11)$$

其中 d_1 为雪崩效应程度的度量; d_2 为完全性程度的度量; d_3 为雪崩效应程度的另一种度量; d_4 为严格雪崩程度的度量. 若 $d_1\approx0.5,d_2=1,d_3\approx1,d_4\approx1$, 则说明算法满足非线性扩散的基本要求, 即可以认为加密函数具有很好的完全性和雪崩效应, 满足严格雪崩准则, 具有很好的混淆扩散特性, 能够抵抗差分密码分析 [33].

本文对 RC5, RC6, 混合混沌分组加密算法 (MCS), AES-Rijndael, SKIPJACK 以及 Logistic 混沌分组加密算法 (即 LCS; LCS 算法是通过单一的 Logistic 方程来实现, 而 MCS 则是通过分段线性映射和猫映射两个混沌方程来实现的, 除此之外 LCS 与 MCS 的设计大体都是相同的) 六种加密算法进行非线性扩散性测试比较, 测试输入向量通过 Matlab 来产生 360000 组 32 bits 的无符号随机整数 [28], 根据不同的算法产生的输出向量有 32 bits, 64 bits 以及 128 bits 三种, 记 $\#T=360000$, 其中 RC5 分组算法的输入输出参数分别为 $n=m=64$ bits, 初始主密钥为随机产生的 256 bits, 迭代轮数为 20(推荐值); RC6 分组算法的输入输出参数分别为 $n=m=128$ bits, 初始主密钥为随机产生的 256 bits, 迭代轮数为 20 (推荐值); MCS 的输入输出参数分别为 $n=m=32$ bits, 初始主密钥为随机产生的 128 bits, 迭代轮数为 13 (推荐值); LCS 的输入输出参数分别为 $n=m=32$ bits, 初始主密钥为随机产生的 128 bits, 迭代轮数与 MCS 相同; AES-Rijndael 算法的输入输出参数分别为 $n=m=128$ bits, 初始主密钥为随机产生的 256 bits, 迭代轮数为 14 (最大值); Skipjack 算法的输入输出参数分别为 $n=m=64$ bits, 初始主密钥为随机产生的 80 bits, 迭代轮数为 32 (推荐值).

表 1 给出了六种加密算法的明文对密文的非线性扩散实验结果; 表 2 给出了六种加密算法的密钥对密文的非线性扩散实验结果. 表 1 与表 2 是在不考虑加密算法的加密模式而只针对算法本身而给出的实验测试结果, 这就更能体现加密算法本身所具有的内在安全性能.

从表 1 和表 2 中可以看出, LCS 大体上是不满足非线性扩散的基本要求的, 而 MCS 是完全满足非线性扩散的基本要求, 具有很好的完全性和雪崩效应, 满足严格雪崩准则, 也即有很好的混淆扩散特性, 能够抵抗差分密码分析攻击, 并且 MCS 的非线性扩散度量值普遍比其他算法要好. 这就说明了混合混沌比单一的 Logistic 混沌在安全性上更具有优势, 同时也表明混沌用于密码学的巨大潜能.

表 1 六种算法的明文对密文的非线性扩散实验结果

测试算法	明文对密文的非线性扩散度量值			
	d_1	d_2	d_3	d_4
RC5	0.500018	1.000000	0.999786	0.999786
RC6	0.499991	1.000000	0. 999779	0.999779
MCS	0.500000	1.000000	0.999809	0. 999809
AES-Rijndael	0.499977	1.000000	0.999781	0.999781
Skipjack	0. 499975	1.000000	0. 999761	0. 999761
LCS	0.470363	0.939453	0.940405	0. 940405

表 2 六种算法的明文对密文的非线性扩散实验结果

测试算法	明文对密文的非线性扩散度量值			
	d_1	d_2	d_3	d_4
RC5	0.500003	1.000000	0.999770	0.999770
RC6	0.500004	1.000000	0. 999799	0.999799
MCS	0.500008	1.000000	0. 999779	0. 999779
AES-Rijndael	0.499997	1.000000	0.999763	0.999763
Skipjack	0.500013	1.000000	0.999765	0.999765
LCS	0.499910	1.000000	0.999626	0.999626

5.2 明文与密文字符数据的平衡性分析

为了验证本文设计的混合混沌加密算法在明文与密文字符的 ASCII 值以及密文的 0-1 二值序列数据平衡性方面的性能, 我们将分别考查明文与密文字符的 ASCII 值的平衡性分布 (图 3 和图 4) 以及密文的 0-1 二值序列数据平衡性分布 (图 5) 两个方面.

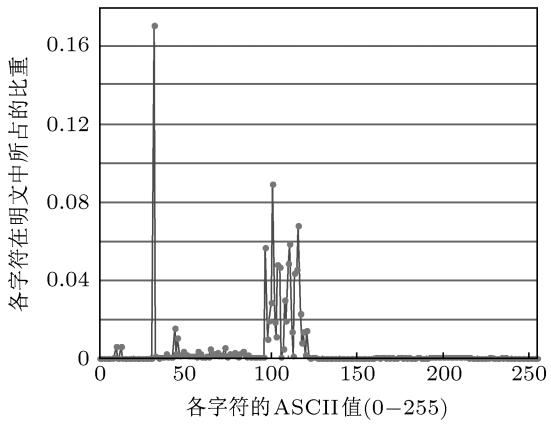

图 3 明文数据中各字符的 ASCII 值占比情况

为了尽可能准确地反映加密算法的真实性能, 本文特选取了一个容量大小为 14.6 MB 的文本文件来进行加密 (CBC 的加密模式), 其中加密初始主密钥 (总共 16 bytes) 随机选取, 同时给定一个大小为 4 bytes 的初始加密向量以及指定加密的轮数为 13.

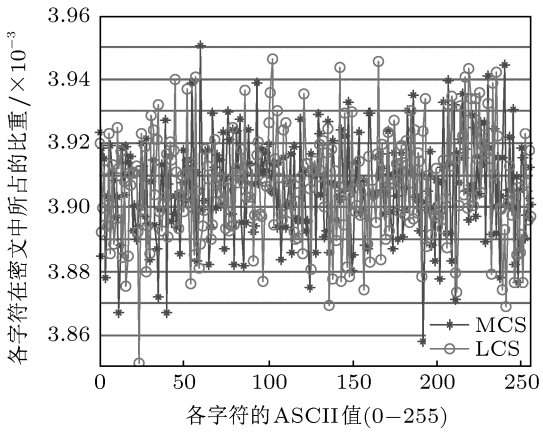

图 4 密文数据中各字符的 ASCII 值占比情况

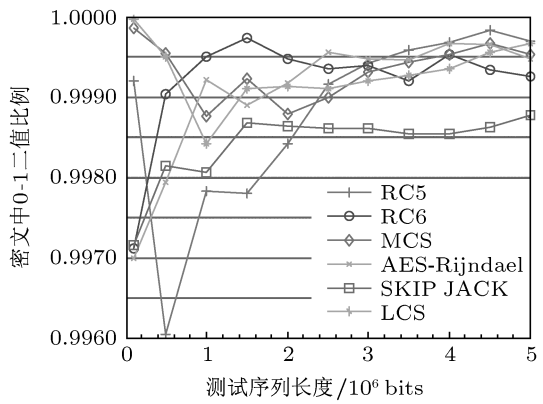

图 5 密文 0-1 二值序列数据平衡性占比情况

从图 3 和图 4 中可以明显地看出, 密文数据与明文数据中各字符的 ASCII 值占比情况有很大的不同: 图 3 中原始明文数据的 ASCII 分布图表明了明文字符的 ASCII 值具有很强的统计学特性, 也即频繁出现在明文字符中的只是很少的一部分字符, 其余大部分字符出现的次数则非常少. 然而理论上要达到实验数据中各字符的 ASCII 值占比应该都集中在 0.00390625 上下范围内, 而图 4 中的密文数据各字符的 ASCII 值占比情况表明了加密后的数据中各字符的占比分布比较均匀, 且其占比都集中在 0.00390625 附近的范围内, 这样就能很好地隐藏变换前的统计分布规律. 在图 4 中给出了 MCS 与 LCS 两种加密算法的实验对比结果, 从中可以看出大部分的点都集中落在 0.00388—0.00393 范围内, 但是 MCS 上的点落在此范围内的点要比 LCS 的多, LCS 上的点落在 0.00385—0.00388 与 0.00393—0.00395 两个范围内的点要比 MCS 多, 这也说明了 MCS 要比 LCS 更具有密码学上的优势.

图 5 是对这六种加密算法输出的密文进行统计性分析得到的 0-1 二值序列数据平衡性占比情况图, 其中纵坐标的 0-1 二值比是通过公式: $1-\frac{|\text{二值}0-1\text{个数之差}|}{\text{密文测试序列长度}}$ 计算得出; 从图 5 中可以看出, 总体上这六种算法的 0-1 二值序列数据平衡性占比都接近 1, 且随着测试序列长度的增大, MCS 的平衡性波动比较小.

5.3 信息熵测试分析

信息熵的公式如 (12) 式所示

$$H(S)=\sum_{S}P(s_i)\log_2\frac{1}{P(s_i)}, \tag{12}$$

$P(s_i)$ 代表着在整个测试信息源 S 中每个字符 s_i 出现的频率, 每个字符占了 8 bits, 如果整个信息源符合统一均匀分布, 则理论上每个字符出现的概率为 1/8, 所以根据 (12) 式累计得到的熵值应该为 8, 所以一个好的加密算法的熵值应该尽可能地接近 8. 从表 3 给出的各种算法加密后得到的熵值可以看出, MCS 得到的熵值要高于其他四种算法, 这说明了 MCS 具有很好的熵测试性能.

表 3 六种算法加密所对应的熵值

熵值	RC5	RC6	MCS	AES-Rijndael	Skipjack	LCS
$H(S)$	7.999982	7.999989	7.999990	7.999987	7.999991	7.999989

5.4 密钥及密钥空间分析

一个好的加密算法不仅要有对密钥的敏感性 (如表 2 所示), 同时还要求该加密算法的内部细节可以公开, 且其安全性完全取决于密钥的安全性, 同时密钥的安全性也是与密钥空间的大小有紧密的联系: 其他条件一定的情况下, 密钥空间越大, 其安全性也就越高, 所以密钥空间应该要大到足以抵抗穷举攻击的程度. 本文设计的混合混沌加密算法的密钥有 128 bits, 同时算法采用 CBC 的加密模式, 所以它的 32 bits 的初始向量可以作为一个辅助密钥, 密钥的组合达到 $2^{160}\approx 1.46\times 10^{48}$ 种可能, 因此以当前的计算能力还暂时无法对该算法成功实施诸如密钥穷举攻击、字典攻击和密钥匹配等类型的强力攻击; 同时混合混沌算法通过了很好的密钥扩展, 从而使得密钥空间中不存在明显的弱密钥以及半弱密钥, 这也在一定程度上增强了该算法的安全性.

5.5 SP 800-22 测试验证

在进行 SP800-22 测试时, 对 122774400 bits 密文数据进行了分析, 其中对这些位的数据分成了 100 组进行测试. MCS 和 LCS 密文测试的综合结果分别如表 4, 表 5 所示.

SP800-22 标准要求在对每一项进行的 100 个测试组中至少要有 96 组通过才能确定通过这项测试; 而在随机漂移和随机漂移变量两项测试中可能发现在给定的 100 个待测试组中只能筛选出 66 (或 70) 组进行测试, 同理标准在对筛选出的能够进行测试的 66 (或 70) 个测试组进行测试时, 要求至少要有 62 (或 66) 组通过才能通过这项测试. 从表 4 和表 5 给出的结果分析可知, MCS 与 LCS 都能通过 SP800-22 标准的各项测试, 然而在与 RC5 的 SP800-22 测试结果比较可以发现 (RC5 的测试结果未给出), RC5 通不过 Overlapping Template 的测试, 这也在一定程度上说明了混沌在密码学领域所具有的优势.

表 4 MCS 密文的 SP800-22 测试

统计测试		比例	P- 值	结果
频率		98/100	0.494392	成功
块频率 ($m = 12280$)		99/100	0.595549	成功
累计总数	Forward	98/100	0.071177	成功
	Reverse	98/100	0.759756	成功
游程		99/100	0.304126	成功
长游程 ($M = 10000$,$N = 100$)		100/100	0.437274	成功
秩		100/100	0.055361	成功
离散 Fourier 变换		100/100	0.304126	成功
不重叠模板匹配 ($m = 9$,$B = 000010011$)		100/100	0.946308	成功
重叠模板匹配 ($m = 9$, $M = 1032$, $N = 968$)		100/100	0.028817	成功
通用统计 ($L = 7$, $Q = 1280$, $K = 141577$)		100/100	0.554420	成功
近似熵 ($m = 10$)		97/100	0.851383	成功
随机漂移 ($x = -3$)		66/66	0.378138	成功
随机漂移变量 ($x = 5$)		65/66	0.964295	成功
串行 ($m = 16$)	P- 值 1	98/100	0.534146	成功
	P- 值 2	99/100	0.678686	成功
线性复杂度 ($M = 1000$)		100/100	0.574903	成功

表 5 LCS 密文的 SP800-22 测试

统计测试		比例	P- 值	结果
频率		99/100	0.304126	成功
块频率 ($m = 12280$)		99/100	0.911413	成功
累计总数	Forward	99/100	0.924076	成功
	Reverse	99/100	0.595549	成功
游程		98/100	0.657933	成功
长游程 ($M = 10000$,$N = 100$)		99/100	0.262249	成功
秩		99/100	0.534146	成功
离散 Fourier 变换		96/100	0.162606	成功
不重叠模板匹配 ($m = 9$,$B = 000010011$)		99/100	0.779188	成功
重叠模板匹配 ($m = 9$, $M = 1032$, $N = 968$)		100/100	0.191687	成功
通用统计 ($L = 7$, $Q = 1280$, $K = 141577$)		97/100	0.759756	成功
近似熵 ($m = 10$)		100/100	0.153763	成功
随机漂移 ($x = -3$)		70/70	0.051001	成功
随机漂移变量 ($x = 5$)		70/70	0.061150	成功
串行 ($m = 16$)	P- 值 1	99/100	0.678686	成功
	P- 值 2	99/100	0.719747	成功
线性复杂度 ($M = 1000$)		98/100	0.994250	成功

5.6 算法执行的速度、时间及存储空间分析

要使混沌加密算法能够用于 WSNS 中, 除了以上安全性要考虑之外, 还必须考虑到其他一些实际的应用性能问题, 这包括执行加密的速度或时间、加密所占的存储空间以及所耗的能量等实际问题. 要说明的是, 其中算法的耗能问题在本文中是通过加密程序运行的时间和速度来反映的, 因为节点耗能问题主要体现在底层通讯、路由和接收发数据等方面. 因为节点资源的限制, 使得 RC6, AES-Rijndael 以及 Skipjack 等算法无法在资源有限的仿真器上直接运行, 所以表 6 仅给出了 MCS, RC5 以及 LCS 三种算法在仿真器上运行的各项测试结果, 其中使用测试的节点仿真器平台是 TI 公司的射频片上系统 CC2430, 它具有以下参数: 一个工业标准增强型的 8051 单片机, 128 K 的 flash, 8 K 的 RAM, 并且其始终频率为 32 MHz. 从表 6 中可以明显地看出, 本文设计的 MCS 无论是在时间、速度还是程序变量所占的空间上都要远远好于 RC5 与 LCS.

表 6 三种算法在节点仿真器上的测试结果

算法 \ 测试项	时间 (μs/byte)	速度(KB/S)	全局变量所占空间 (byte)	局部变量所占空间 (byte)
MCS	725	1.347	16	18—20
RC5	876	1.115	168	88—102
LCS	790	1.236	26	25—41

6 结 论

传统的加密方法如 DES, AES, RSA 等因其所需的资源多、能耗大和速度慢而不适用于 WSNS 节点的数据加密, 从而导致能够用于 WSNS 节点的加密算法少. 针对这种情况, 本文给出了一种将两个典型的混沌方程 (分段线性映射和猫映射) 分别进行数字化的方法, 并且通过结合 Feistel 网络结构设计了一种新的分组加密算法 MCS. 通过对其进行大量的实验测试, 结果表明该算法具有密钥空间大、严格的雪崩效应、非常好的扩散和扰乱性能以及均匀的统计平衡性, 并通过 SP 800-22 测试验证了该算法; 同时在仿真器平台上进行的速度、时间及存储空间分析与测试表明了该算法能够很好地用于 WSNS 中节点的数据加密.

这种新的分组加密算法通过与 RC5, RC6, AES-Rijindael 以及 SKIPJACK 等常用于 WSNS 中的几种算法进行了比较之后, 发现其在安全性、保密性、耗能性、占用资源以及速度或时间等方面都比其他几种加密算法更具优势, 原因是由于混沌具有了密码学所要求的特性, 并且这种新的分组加密算法采用了 Feistel 结构、基于加密轮数的动态迭代技术以及改进的密钥扩展算法; 另外在与 LCS 算法进行比较时发现, 由于新的分组加密算法使用了混合混沌加密技术, 从而使得其在安全性方面要远远高于 LCS 算法.

本文设计的新分组加密算法是一种速度快、安全性高、保密性强且资源消耗低、适用于 WSNS 节点的算法, 它为加快 WSNS 安全机制的研究提供了理论基础.

[1] Tilak S, Ghazaleh N B, Heinzelman W 2002 *Mob. Comput. Commun. Rev.* **1** 1

[2] Sun L M, Li J Z, Chen Y, Zhu H S 2005 *Wireless Sensor Networks* (Beijing: Tsinghua University Press) p37(in Chinese) [孙利民, 李建中, 陈渝, 朱红松 2005 无线传感器网络 (北京: 清华大学出版社) 第 37 页]

[3] Yang J Y 2007 *Ph. D. Dissertation* (Chongqing: Chongqing University) (in Chinese) [杨吉云 2007 博士学位论文 (重庆: 重庆大学)]

[4] Wang S 2007 *The Theory and Application for Wireless Sensor Networks* (Beijing: Beihang University Press) pp7—9 (in Chinese) [王殊 2007 无线传感器网络的理论及应用 (北京: 北京航空航天大学出版社) 第 7–9 页]

[5] Liu S D, Liang F M, Liu S K 2003 *The Chaos and Fractal in Nature Science* (Beijing: Beijing University Press) p26 (in Chinese) [刘式达, 梁福明, 刘式适 2003 自然科学中的混沌和分形 (北京: 北京大学出版社) 第 26 页]

[6] Liao X F, Xiao D, Chen Y 2009 *The Principle and Application of Chaotic Cryptography* (Beijing: Science Press) p1 (in Chinese) [廖晓峰, 肖迪, 陈永 2009 混沌密码学原理及其应用 (北京: 科学出版社) 第 1 页]

[7] Xu S J, Wang J Z 2008 *Acta Phys. Sin.* **57** 37 (in Chinese) [徐淑奖, 王继志 2008 物理学报 **57** 37]

[8] Xie K, Lei M, Feng Z J 2005 *Acta Phys. Sin.* **54** 1267 (in Chinese) [谢鲲, 雷敏, 冯正进 2005 物理学报 **54** 1267].

[9] Liu M H, Feng J C 2009 *Acta Phys. Sin.* **58** 4457 (in Chinese) [刘明华, 冯久超 2009 物理学报 **58** 4457]

[10] Li J B 2011 *Acta Phys. Sin.* **60** 060508 (in Chinese) [李家标 2011 物理学报 **60** 060508]

[11] Gu Q L, Gao T G 2009 *Chin. Phys.* B **18** 84

[12] Hu J F, Guo J B 2008 *Acta Phys. Sin.* **57** 1477 (in Chinese) [胡进峰, 郭静波 2008 物理学报 **57** 1477]

[13] Li W, Hao J H, Qi B 2008 *Acta Phys. Sin.* **57** 1398 (in Chinese) [李伟, 郝建红, 祁兵 2008 物理学报 **57** 1398]

[14] Ji J X, Qiu S S 2010 *Acta Phys. Sin.* **59** (in Chinese) [晋建秀, 丘水生 2010 物理学报 **59** 792]

[15] Liu S B, Sun Q, Xu Z Q, Liu J S 2009 *Chin. Phys.* B **18** 5219

[16] Chen S 2006 *Ph. D. Dissertation* (Chongqing: Chongqing University) (in Chinese) [陈帅 2006 博士学位论文 (重庆: 重庆大学)]

[17] Tan Y J 2010 *Ph. D. Dissertation* (Chengdu: Electric and Scientific University) (in Chinese) [谭益军 2010 博士学位论文 (成都: 电子科技大学)]

[18] Rivest R L 1995 *Dr. Dobb's Journal* **20** 146

[19] Rivest R L 1995 *Lecture Notes in Computer Science* **1008** 86

[20] Yee W L, Jeroen D, Pieter H 2006 *ACM Transactions on Sensor Networks (TOSN)* **2** 65

[21] Rivest R L 1994 *Proceedings of the Second International Workshop on Fast Software Encryption* (New York: Springer-Verlag) p86

[22] Guido B, Luca B, Israel K Paolo M, Vincenzo P 2003 *14th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP*'03) Hague, Netherlands, 24–26 June, 2003 p423

[23] Rivest R L Http://people.csail.mit.edu/rivest/RC6.pdf 1998-08-20
[24] Sun S L 2004 *Application Cryptography* (Beijing: Tsinghua University press) p(23) (in Chinese) [孙淑玲 2004 应用密码学 (北京: 清华大学出版社) 第 23 页]
[25] Advanced Encrtption Standard.buchholz.hs-bremen.de/aes/AES.pdf 2006-02-20
[26] Advanced Encryption Standard Home Page http://csrc.nist.gov/publications/fips/ fips197/ fips-197.pdf 2001-11
[27] National Institute of Standards and Technology http://csrc.nist.gov/groups/ST/toolkit/ documents/ skipjack/skipjack.pdf 1994-2
[28] Peng J , Liao X F 2006 *J. Electron. Inform.* **28** 4 (in Chinese) [彭军, 廖晓峰 2006 电子与信息学报 **28** 4]
[29] Mohammad P, Nevenko Z 2000 *Comput. Secur.* **19** 467
[30] Liu Y Z 2005 *Cryptography and Network Security—Principle and Practice* p135 (in Chinese) [刘玉珍 2005 密码编码学与网络安全 —— 原理与实践 (电子工业出版社) 第 135 页]
[31] Shannon C E 1949 *The Bell Syst. Tech. J.* **28** 656
[32] Preneel B, Bosselaers A, Rijmen V http://www.nist.gov /aes 2000-05
[33] Zhu M F, Zhang B D, Lv S W 2002 *J. Commun.* **23** 10 [朱明富, 张宝东, 吕述望 2002 通信学报 **23** 10]

The novel block encryption scheme based on hybrid chaotic maps for the wireless sensor networks*

Tong Xiao-Jun† Zuo Ke Wang Zhu

(*School of Computer Science and Technology, Harbin Institute of Technology, Weihai* 264209*, China*)

(Received 6 May 2011; revised manuscript received 9 June 2011)

Abstract

Traditional encryption schemes are not suitable for the Wireless sensor networks (WSNS) due to some intrinsic features of nodes in WSNS such as low energy, limited computation ability and storage resources. In this paper, we present a novel block encryption scheme based on hybrid chaotic maps dynamically and propose an integer digital random method, and the Feistel network structure, which is a kind of fast, secure, low resource consumption and suited for WSNS nodes encryption scheme. The experimental tests show the new encryption scheme has the following prefect performances: large key space, very good diffusion and disrupt performance, strict avalanche effect, excellent statistical balance and fast encryption speed of the new scheme, and the encryption scheme passes the SP800-22 test; meanwhile, the analysis and the testing of speed, time and storage space on the simulator platform show that this new encryption scheme is well able to hide the data information about the node in WSNS.

Keywords: WSNS, hybrid chaos, block algorithm, data encryption

PACS: 05.45.+b

* Project supported by the National Natural Science Foundation of China (Grant No. 60973162), the Natural Science Foundation of Shandong Province of China (Grant No. ZR2009GM037), the Science and Technology of Shandong Province of China (Grant No. 2010GGX10132), the Key Natural Science Foundation of Shandong Province of China (Grant No. Z2006G01), the Scientific Research Foundation of Harbin Institute of Technology at Weihai, China (Grant No. HIT(WH) ZB200909), the Technology Research and Development Program of Weihai High-Tech Development Zone in Shandong Province of China (Grant No. 201025), and the Technology Research and Development Program of Weihai, China (Grant No. 2008011).

† E-mail: tong_xiaojun@163.com