doi:10.3969/j.issn.1000-436x.2013.05.013

面向无线传感器网络的混沌加密与消息鉴别算法

陈铁明[1,2]，葛亮[1]

（1. 浙江工业大学 计算机科学与技术学院，浙江 杭州 310023；
2. 北京航空航天大学 软件开发环境国家重点实验室，北京 100191）

摘 要：针对数字混沌密码无法直接在轻量的传感节点上实现，介绍了一种基于整数型混沌映射的轻量级分组加密算法，并提出一种新型的消息鉴别码方案，具有输出长度可变、混沌序列复合安全等特点，最后实现了一个完整的无线传感网数据安全传输应用方案，并在 Mica2 节点上给出性能分析。实验表明该方案对无线传感网具有更高的实用性。

关键词：无线传感器网络；混沌映射；分组加密；散列函数；消息鉴别码

中图分类号：TP309 **文献标识码**：B **文章编号**：1000-436X(2013)05-0113-08

Chaos-based encryption and message authentication algorithm for wireless sensor network

CHEN Tie-ming[1,2], GE Liang[1]

(1. College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China;
2. State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China)

Abstract: As for the traditional digital chaos-based scheme being not directly applicable for WSN because some sensor node does not support the float type of computation, integer chaotic maps and its block encryption schemes are firstly introduced, and then a novel chaos-based message authentication code (MAC) scheme was proposed, which was length changeable with high-level hybrid chaotic security. At last, a security application solution for data communication on WSN employing the proposed chaos-based schemes is presented, and the performance evaluation on the real-world Mica2 node is analyzed to show that the new security solution is more applicable for WSN than the current most popular security scheme, TinySec.

Key words: wireless sensor network; chaos mapping; block encryption; hash function; MAC

1 引言

随着无线传感器网络(WSN, wireless sensor network)的快速发展，面向 WSN 的密码研究引起了广泛关注。尽管各种公钥算法应用在 WSN 上的研究已获得较多成果[1~3]，但由于传感节点的计算和存储资源有限，目前，面向 WSN 的密码应用大多采用对称加密算法，如 RC5、RC6、SKIPJACK 和 AES 等。RC5[4]是 Rivest 设计的一种分组加密算法，运算过程无需存储大容量中间表，仅采用加、减、异或和移位等基本运算，较适合于 WSN 节点，但 RC5 是 RSA 公司专利产品，在实际使用时受限制；RC6[5]是在 RC5 基础上提出的一种 AES 候选算法，基本保留了 RC5 运算简单、无需大容量表存储的优

收稿日期：2012-06-26；**修回日期**：2013-03-21

基金项目：国家自然科学基金资助项目（61103044）；浙江省自然科学基金资助项目（Y1110576）；浙江省科技厅公益技术研究计划基金资助项目（2011C21046）；“十二五”国家密码发展基金密码理论课题基金资助项目（MMJJ201101009）；北京航空航天大学软件开发环境国家重点实验室开放课题基金资助项目（SKLSDE-2011KF-07）

Foundation Items: The National Natural Science Foundation of China (61103044); The Natural Science Foundation of Zhejiang (Y1110576); Zhejiang Public Technology Research Project (2011C21046); The National Cryptography Development Foundation of China(MMJJ201101009); The State Key Laboratory of Software Development and Environment of Beihang University (SKLSDE-2011KF-07)

点，且算法免费，但 RC6 算法的最小分组长度为 128bit，针对小数据通信量的 WSN 应用必然会产生较多冗余数据，因而造成传感节点能量的浪费；SKIPJACK[6]是目前被 WSN 广泛应用的一个分组加密算法，虽然分组长度仅 64bit，较适用于 WSN 加密通信，但需存储中间计算所需的轮函数表，耗费节点资源，且 WSN 软件实现的性能有待提高。另外，鉴于 Rijndael 算法(AES)是目前的分组对称加密新标准，IEEE 802.15.4 已提出基于 AES 的加密方案，但 AES 算法采用代替/置换网络，需较大容量的表来存储 S 盒，因而也并不适合在 WSN 节点上的软件实现[7]。

因此，适用于 WSN 节点运算的新型轻量级密码算法成为当前的一个研究热点[8]。混沌密码一种应用较为广泛的非传统密码技术，将混沌密码应用于 WSN 已开始得到较多关注[9]。最近，有学者提出了整数型混沌迭代运算方法，并提出了一种具有 Feistel 结构的混沌分组密码，分组长度仅为 8bit，可有效避免因填充而造成的数据冗余问题，适用于数据分组长度较短的 WSN 网络通信[10]。

本文的工作即将在研究基于整数型混沌映射的分组加密算法基础上[10]，提出一种基于混沌分组加密的轻量级消息鉴别码方案，并给出与基于混沌的一般散列算法[11,12]以及 MD5 算法的性能优势分析，最后设计实现一个 WSN 数据安全应用协议方案，并在 Mica2 节点上给出性能测试，实验结果表明利用本文提出的消息鉴别算法构建的安全协议性能优于 TinySec。

2 面向 WSN 的混沌分组加密模式

2.1 混沌映射的整数化方法

无线传感器节点通常采用嵌入式处理器，如 Mica2 节点使用 8bit 的 ATmega128L 处理器芯片，一般都无法直接支持浮点数等复杂运算，而传统的混沌映射系统的工作域通常为连续实数域。因此，为了使混沌密码在无线传感嵌入式节点上实现，首先需采用离散化的整数型混沌映射方法。

本文选取最为常用的 Logistic 一维映射，其混沌函数如下

$$x_{n+1}=\mu x_n(1-x_n) \tag{1}$$

其中，x_n 是第 n 次迭代的结果，x_{n+1} 代表第 $n+1$ 次的迭代结果，μ 是系统参数。当 μ 处于 $(3.57,4]$ 区段时，Logistics 映射呈现混沌特性，系统迭代值的分布从统计学上来看具有类似白噪声的性质，即混沌特点[13]。

尽管 Logistic 映射在时域上离散，但在值域上仍是连续的。因此，根据文献[10]中的方法，可将其改造成一种时域和值域均离散化的整数型混沌映射。

对于如式（1）的 Logistic 映射，存在一种等价的方式如下

$$x_{n+1}=1-\lambda x_n^2 \tag{2}$$

其中，$\lambda\in[0,2], x_n\in[-1,1]$。式(2)两边乘上 $a^2(a\neq 0)$，得到

$$a^2x_{n+1}=a^2-\lambda(ax_n)^2 \tag{3}$$

令 $z_n=ax_n+a$，则

$$\begin{cases} x_n=\dfrac{z_n}{a}-1 \\ x_{n+1}=\dfrac{z_{n+1}}{a}-1 \end{cases} \tag{4}$$

将式（4）代入到式（3）中，取 $\lambda=2$，化简即得

$$z_{n+1}=4z_n-\frac{2}{a}z_n^2 \tag{5}$$

因为 $x_n\in[-1,1]$，所以 $z_n\in[0,2a]$。令 z_n 的取值全部为整数，若取 $a=2^{L-1}$，L 为机器字长，则 z_n 值正好是机器字长所能表示的整个无符号整数范围，式（5）即为机器字长表示的无符号整数范围内的迭代运算式。

对于如式（5）的迭代运算，容易在嵌入式系统中计算得到 z_n 的值。例如，若机器字长为 16bit，则可取 $a=2^{16}/2=2^{15}=32\,768$，此时的 $z_n\in[0,65\,536]$ 正好对应于 16bit 所能表示的无符号整数范围。在迭代的过程中，计算 $4z_n$ 时仅需将 z_n 左移两位；计算 $\frac{2}{a}z_n^2=\frac{2}{32\,768}z_n^2=z_n^2\times 2^{-14}$ 时仅需将 z_n^2 右移 14bit。因此，对于式（5）的迭代运算，整个计算的过程仅需整数加/减法、乘法和移位操作，适合于 WSN 节点性能有限的嵌入式芯片。实验表明，当字长选择 32bit 时，式(5)的迭代序列已表现出较理想的随机特性，可在 WSN 安全强度和计算性能方面获得较好的平衡。因此，本文后续的密码设计就将采用 32bit 的整数型混沌序列（NesC 语言中的 uint32_t 类型表示迭代值）。

2.2　基于 Feistel 结构的分组加密算法

2.2.1　8bit Feistel 加密结构

分组长度为 8bit 的 Feistel 结构如图 1 所示。明文的一个分组被分为高位和低位各 4bit，分别记为 L_i 和 R_i。在轮密钥 k_i 的作用下，R_{i-1} 通过 f 函数后与 L_{i-1} 异或生成新的 R_i，而 R_{i-1} 变为新的 L_i，由此完成一轮 Feistel 运算。

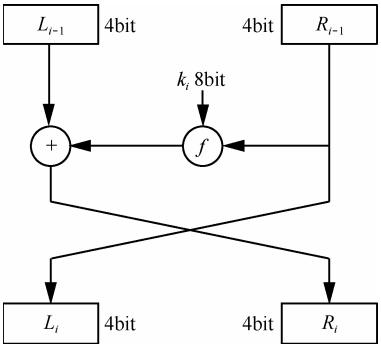


图 1　8bit 分组的 Feistel 结构

图 1 所示的 Feistel 结构可表示为

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus F \end{cases} \tag{6}$$

其中，R_i、R_{i-1}、L_i、L_{i-1}、F 的长度均为 4bit。

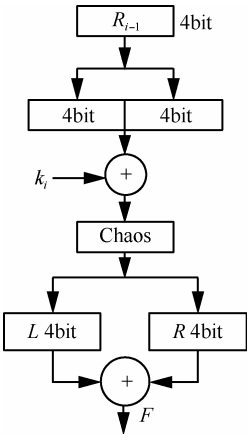


图 2　Feistel 结构中的 f 函数

f 函数为 8bit 的整数型混沌迭代式，其结构如图 2 所示。低 4bit 的 R_{i-1} 首先被扩充到 8bit，与同样 8bit 的轮密钥 k_i 异或后，输入到 Chaos 函数中进行 8bit 整数型混沌迭代，输出的 8bit 迭代值再次分为高低各 4bit，相互异或后生成最终的输出 F。

Chaos 函数接收输入作为 z_n，迭代后输出为 z_{n+1}。每一轮的中间输出 F 与输入的 R_{i-1} 及当前的轮密钥 k_i 相关，且利用了 8bit 整数混沌计算的非线性，提高了加密轮的安全性。

2.2.2　加解密过程

完整的 8bit 分组加密过程如图 3 所示。在明文分组进入 Feistel 加密之前，对 8bit 数据进行 P 置换。加密过程中的轮数可变，同一般的 Feistel 结构分组加密一样，最后一轮的输出不作左右交互。最终的 8bit 输出进行逆 P 置换作为加密结果。

Feistel 结构可使加密过程与解密过程具有完全相同的结构，仅需逆序地使用加密的轮密钥即可完成解密操作。例如若加密时采用了 4 轮子密钥 k_1、k_2、k_3、k_4，则解密时正确的子密钥顺序为 k_4、k_3、k_2、k_1。

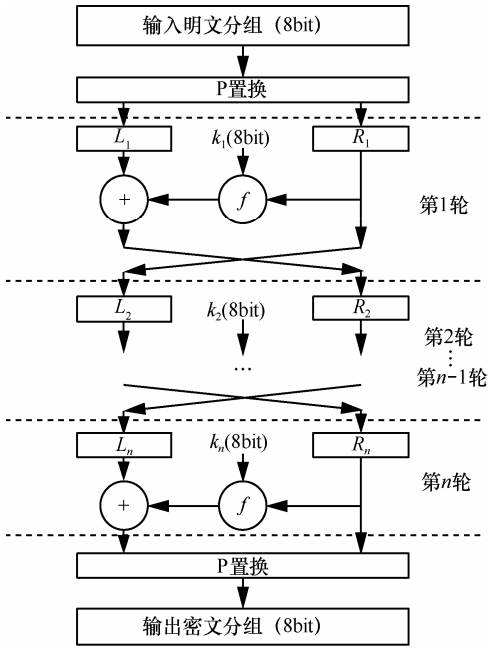


图 3　8bit WSN 分组加密流程

2.2.3　安全性能分析

加密过程中，由于将整数型混沌序列作为轮加密的子密钥，增强了系统安全性，且通过增加 Feistel 结构的轮数，同时也增加了子密钥数量，进一步提高了加密安全性。

本文选取 4 轮加密，记子密钥为 k_1、k_2、k_3、k_4。每个子密钥的长度均为 8bit，完成单个分组的一次加密或解密一共仅需 32bit 的密钥，恰好为本文 2.1 节讨论的 32bit 整数型混沌序列的一次迭代值。因此，上述 8bit 的分组加密模式实现了一次混沌映射迭代加密一个明文分组，即在保证密码系统安全的同时，保障了较高的加密执行效率。

另外，AES、SkipJack 等经典加密算法采用了大量置换表，在软件实现算法时存在较大的硬件存储需求，而上述的 8bit 分组加密过程的置换结构简单，无需复杂的置换表，适用于存储和计算均受限的 WSN 节点。

3 基于混沌分组加密的 WSN 消息鉴别码

3.1 消息鉴别算法设计

用 CB 代表上述基于 Feistel 结构的 8bit 分组混沌加密函数，在此基础上利用密文反馈链（CBC）模式构造一个新型混沌散列函数，用于计算消息的摘要，即消息鉴别码（MAC）。消息鉴别算法的结构如图 4 所示，将原始消息 M 分组为单字节的 $M_1,M_2,\cdots,M_n$。

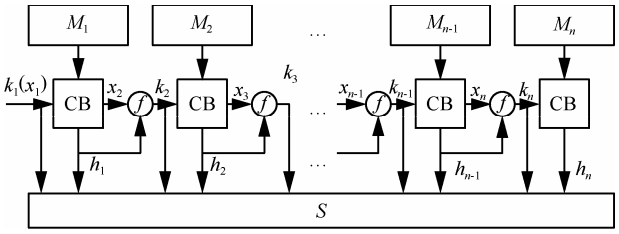


图 4 基于 8bit 混沌分组加密结构的消息鉴别算法框架

算法仍采用 32bit 整数型 Logistic 混沌映射，图 4 中 $x_1,x_2,\cdots,x_n$ 为映射迭代值。在初始密钥 k_1 的作用下，M_1 被加密成与其等长的 $h_1=CB_{k_1}(M_1)$，k_1 和 h_1 作为 S 函数的输入参数，计算最终的输出。同时，h_1 与新的迭代值 x_2 在 f 函数作用下生成下一分组所需的密钥 k_2。重复上述步骤，直到处理完所有消息分组。最终 MAC 码的字节长度记为 b，用字节数组 CMAC[b]表示。

图 4 中的 S 函数结构如图 5 所示，h 的长度为 1byte，CMAC 的长度为 b byte。这里考虑将 h 的每一位扩散到这 b byte 单元的对应比特上，将输入值 k 作为索引值 $Index$，用于确定目标单元在数组 $CMAC[b]$ 中的下标。

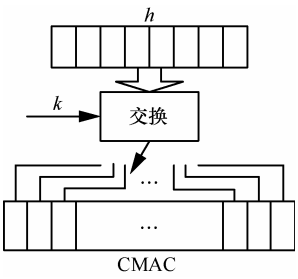


图 5 混沌消息鉴别算法中的 S 函数结构

如下定义 h_i：将输入值 h 的第 i 比特数据保留，其余比特置零。

S 函数为每一个分组消息都执行一次循环计算操作如下。

遍历 i（从 1 到 8）。

1) k 循环左移 $i-1$ 比特，截取 k 的前 l 比特构成整数 $Index$。其中，l 的值和 $CMAC$ 长度 b 之间满足

$$2^l=b \tag{7}$$

2) 以 $Index$ 值作为下标，进行如下操作

$$CMAC[Index]=CMAC[Index]\oplus h_i \tag{8}$$

图 4 中的 f 函数定义为

$$k_n=x_n+h_{n-1}\bmod x_{\max}=L(k_{n-1})+h_{n-1}\bmod x_{\max} \tag{9}$$

这里 $k_1=x_1$，L 为形如式（5）的混沌映射，$x_{\max}$ 为映射过程中的迭代值上限。

上述 S 函数、f 函数以及 CBC 模式的混合使用加强了整个散列函数的扩散效应。

3.2 安全及性能分析

目前，常用的散列函数如 MD5，产生的输出为 128bit，即 16byte。为了便于分析对比本文提出的混沌散列函数与现有散列函数的性能，将长度设置成 16byte，即式（7）中的 $b=16,l=4$。

随机选取 10kbyte 长度的消息作为测试样本。本文选用了 Windows XP 系统更新日志记录文本 WindowsUpdate.log，该文本文件记录了系统最近 1 个月的操作日志。

下面分几种情况分别计算日志消息的 MAC 码，并进行比较。

Case 1 计算原始消息的 MAC 码。

Case 2 将原消息中第一个数字“2”改成“3”，计算更改后的消息 MAC 码。

Case 3 在原消息中的“Shutdwn”中加入“o”，即改成“Shutdown”，计算更改后的消息 MAC 码。

Case 4 将原消息中的“health”改成“error”，计算更改后的消息 MAC 码。

Case 5 去除原消息中第一行末尾的“.”符号，计算更改后的消息 MAC 码。

Case 6 交换原消息中的“event”和“state”，计算更改后的消息 MAC 码。

计算得到相应的 MAC 码的十六进制表示如下。

Case 1

D4410C3237715AA8584E2471E4BE15B2

Case 2

98B83FC0D7892E332B3DEE52C15D4521

Case 3

558A11AD83BAE86599EA05B16AF79271

Case 4

EA5AAC947DA8F062B48F9E6E5D1D0493

Case 5

F9CA68EC57FBF5DC7E6C541C50CF3486

Case 6

59809E6B99D59975428AEA3FC0123A84

以上的测试结果表明，本文提出的算法满足散列函数的最基本属性，即对消息压缩摘要的同时，尽可能地将原文信息扩散到输出的每一比特当中；任何明文中的细微修改都会导致函数输出 MAC 码的大量改变，实现雪崩效应。

下面对本文提出的消息鉴别算法进一步给出各项性能测试。

1) 统计测试

任意选取 10kbyte 长度的消息，计算其散列值，记为 $cmac_1$。翻转（toggle）消息中的任意一比特信息，计算新的消息的散列值，记为 $cmac_2$，统计 $cmac_2$ 和 $cmac_1$ 相比改变的比特数，记为 B。重复该测试 N 次，N=1 024 时的结果如图 6 和图 7 所示。

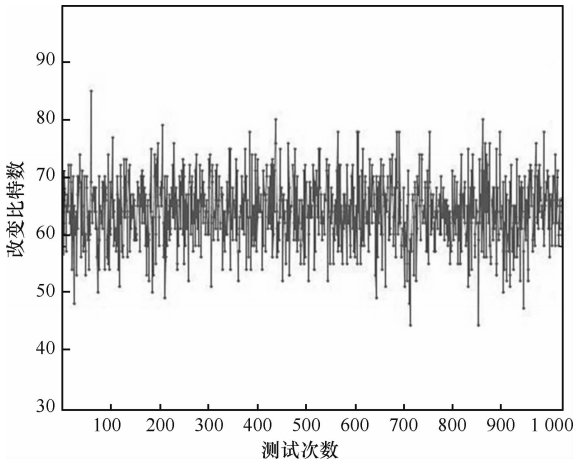


图 6　MAC 码改变比特数统计

理想的扩散和混乱可使消息发生微小变化时，其散列值的每一比特都获得 50%的几率发生改变。实验结果表明，散列函数计算得到的 MAC 码长度为 128bit，而每次测试均有大约 55~75bit 发生了改变，且这些数值大多分布在 64 附近，与理想情况接近。

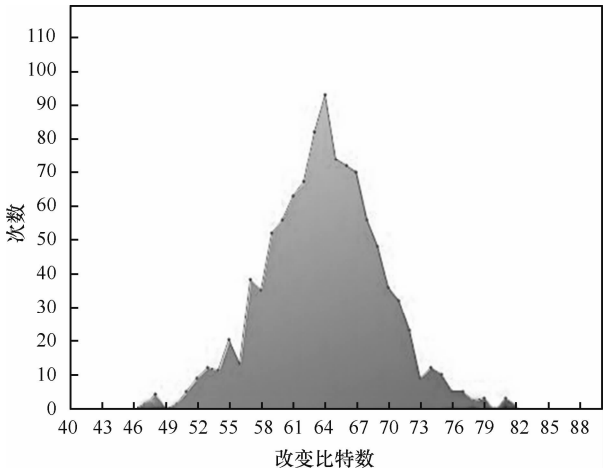


图 7　MAC 码改变位数分布统计

以下是另外一些常用的散列函数性能测试指标

$$\overline{B}=\frac{1}{N}\sum_{i=1}^{N}B_i \tag{10}$$

$$P=(\overline{B}/128)\times 100\% \tag{11}$$

$$\Delta B=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i-\overline{B})^2} \tag{12}$$

$$\Delta P=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i/128-P)^2}\times 100\% \tag{13}$$

其中，式（10）和式（11）计算改变比特数（率）的均值，而式（12）和式（13）的标准差值则体现了样本值与均值的偏差程度，值越小说明扩散和混乱特性越明显。

对于 N=256、512、1 024、2 048，分别计算式（10）~式（13）的结果，与现有几种算法进行比较，结果如表 1 所示。其中，Wang[11]和 Zhang[12]提出的是 2 种当前较典型的混沌散列函数，MD5 则代表被广泛采用的传统散列算法。

由分析结果知，上述 3 种混沌散列算法在统计测试中的表现均十分突出，而本方案在多数情况下的统计测试结果均优于 MD5 算法。

2) 碰撞测试

任意选取 10kbyte 长度的消息，计算其散列值，记为 $cmac_1$。翻转（toggle）消息中的任意一比特信息，计算得到 $cmac_2$，统计 $cmac_1$ 和 $cmac_2$ 中对应位置上相同的比特数，记为 *hit* 。重复该测试 10 000 次，得到实验结果如图 8 所示。分析结果表明，对应位置上具有 3 个相同比特的情况为 0，且只有极少数情况出现 2 个相同，表现出极高的抗碰撞性。

表 1 散列算法各类统计测试结果对比

N	$\overline{B}$				P/%				ΔB				ΔP/%			
	本方案	Wang[11]	Zhang[12]	MD5	本方案	Wang[11]	Zhang[12]	MD5	本方案	Wang[11]	Zhang[12]	MD5	本方案	Wang[11]	Zhang[12]	MD5
N=256	63.98	64.09	63.35	63.68	49.99	50.07	49.32	49.75	5.48	5.38	5.93	5.38	4.28	4.21	5.01	4.20
N=512	63.89	63.92	64.62	63.92	49.91	49.94	50.53	49.93	5.60	5.62	5.82	5.78	4.38	4.39	4.93	4.36
N=1 024	63.97	64.08	63.41	63.98	49.98	50.06	49.49	49.98	5.65	5.56	5.68	5.73	4.41	4.34	4.68	4.48
N=2 048	64.03	63.98	64.63	64.03	50.03	49.98	49.46	50.02	5.61	5.53	5.57	5.66	4.38	4.33	4.51	4.42

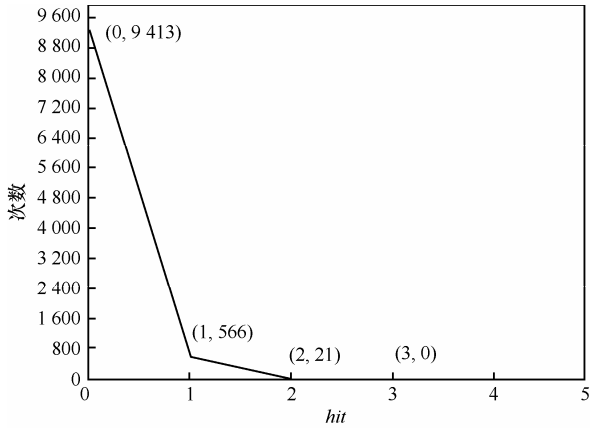


图 8 碰撞测试（*hit* 数）统计图

另外，表 2 进一步给出了几种算法的碰撞测试结果比较，本文方案优于 Zhang 方案。

表 2 碰撞测试（最大 ***hit*** 数）统计结果对比

方案	碰撞 *hit* 数
本方案	2
Wang[11]	2
Zhang[12]	3
MD5	2

另一个常用的碰撞测试指标如下：记 MAC 码每个字节为 e_i，$t(e_i)$ 为该 ASCII 字符的数值，计算绝对差 d

$$d=\sum_{i=1}^{N}\left|t(e_i)-t(e_i^{'})\right| \tag{14}$$

表 3 给出了几种算法的绝对差测试结果比较，本文方案优于 Wang 方案。

表 3 碰撞测试（绝对差）统计结果对比

方案	绝对差 d		
	最大值	最小值	平均
本方案	2 283	644	1 372
Wang[11]	2 295	689	1 526
Zhang[12]	2 014	546	1 350
MD5	2 074	590	1 304

3) 实用性测试

针对 WSN 节点，计算 128bit MAC 码除了耗费内存资源外，还将产生过大的数据传输量，因为在 WSN 中，通常节点一次发送的消息大小仅为 10~20byte，若 MAC 码为 16byte，则数据分组有效载荷仅在一半以下，效率太低、能耗过大。因此，128bit 长度的鉴别码并不适合 WSN 应用。本文提出的混沌散列函数的输出长度可变，仅需改变参数 b 的值，即可构造不同输出长度的消息认证码。从理论上分析，过短的散列函数发生碰撞的可能性会增高，因此需要权衡考虑算法性能和应用可行性。参照目前被 TinyOS 集成的安全协议 TinySec[14]采用的 MAC 码长度，将 b 减小为 4，即散列函数的输出长度为 32bit。

对 32bit 的散列函数重新进行各项测试，结果如表 4 和表 5 所示。分析结果表明，尽管鉴别码长度减小，但散列函数仍具有较好的统计特性，如混淆和扩散仍接近 50%，字节碰撞测试的绝对值仍达到一定水平，可满足 WSN 一般应用的数据安全强度。

表 4 32bit 输出长度的统计测试结果对比

N	$\overline{B}$	P/%	ΔB	ΔP/%
N=256	16.18	50.54	2.86	8.94
N=512	15.88	49.64	2.85	8.92
N=1 024	16.04	50.11	2.81	8.79
N=2 048	16.17	50.54	2.79	8.74

表 5 32bit 输出长度的碰撞测试（绝对差）

绝对差 d	本方案
最大值	553
最小值	84
平均	288

4 WSN 数据安全应用方案

4.1 安全应用设计

下面设计并实现一个基于上述混沌分组加密和消息鉴别算法的数据安全传输应用协议，保障

WSN 数据传输过程中的保密性和完整性。

演示采用 Crossbow 公司的 Mica2 系列节点以及数据采集板 MTS300 等装置搭建一个由基站和传感器节点组成的环境信息采集系统，完成实验室环境信息的实时采集与显示。传感器网络监测实验室分布区域的温度、光照等，通过短距离射频通信传给网关/汇聚节点（Node 0），由网关节点连接终端的计算机，通过串口，用户可以观察到节点监测区域的信息。节点采用 Mica2 节点，节点上配有 CC1000 短距离无线通信芯片以及 ATmega128L 处理器芯片。其中，0 号节点连接到 MIB520 USB 接口板，共同组成一个基站并连接到计算机。监测节点布置在待监测区域内，且节点通过 51 针接口连接 MTS310 传感器板。传感器板上配有温度传感器（panasonic ERT-J1VR103J）、光照传感器硒化镉（CdSe）光电池等。传感器采集到的信息经过 ADC 数模转换后由处理器进行处理，通过 CC1000 经由天线发送到 0 号节点即基站。在 TinyOS 原有的通信协议栈中，提供了一组通用的数据通信组件 GenericComm/ AMStandard，该组件对上层应用提供了 SendMsg/ ReceiveMsg 接口，接受应用程序的数据发送请求，同时响应并传递数据接收事件。在实际工作时，根据不同的硬件平台，GenericComm/ AM-Standard 组件将和对应的组件导通，使用其接口实现通信功能。例如，对于本实验使用的 Mica2 节点，这里的导通组件即为 CC1000RadioIntM 组件。

将混沌密码系统加入到 CC1000RadioIntM 组件中，节点利用该组件收到应用层数据分组并在发送之前，对数据分组净载荷部分进行加密，之后计算整个数据分组的消息认证码（MAC），将其连同原数据分组一起发送；基站接收到数据分组后，首先进行 MAC 码的验证，并进行随后的解密操作，这些步骤完成后，数据分组才被传递给上层应用。整个方案为无线传感器网络数据传输提供了更好的安全性，同时体现了对应用程序的透明性。

数据加密使用 8bit 长度的混沌分组加密算法，由于算法的分组长度为 8bit，即单个字节，有效避免了因分组过长而加入过多冗余数据，减少了 WSN 需要传输的数据量，节约了能量资源和通信信道资源；消息鉴别码 MAC 采用本文提出的 32bit 混沌散列算法，与 TinyOS 原始数据报文中的冗余码 CRC 保持长度一致。图 9 给出了 TinyOS 系统中数据报文加密前后的格式对比。

Dest	AM	L	G	Data(default length: 29byte)	CRC

(a) TinyOS原始数据报文格式

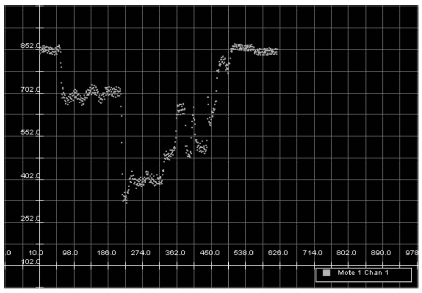
Dest	AM	L	G	Data(default length: 29byte)	MAC

(b) TinyOS安全数据报文格式

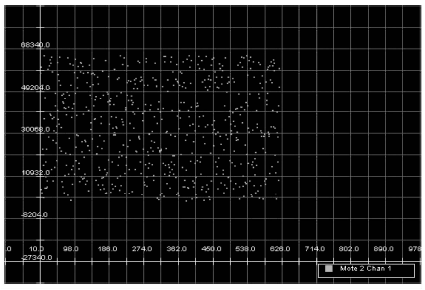
图 9　加密前后 TinyOS 数据分组格式对比

4.2　安全应用测试

基站持续接收节点发送的数据分组，解密并提取出其中的环境光照强度采样值，绘制出如图 10(a)的环境光强变化图，横坐标为采样值编号，纵坐标为光照强度值，图中清楚地显示了监测区域的光照强度变化，图 10(b)则为数据解密错误时的光照强度变化图。由实验结果知，用错误的密钥解密得到的数据散乱分布在整个取值空间上，且从图中完全无法看出数据的变化趋势。对比图 10(a)和图 10(b)，利用本文提出的数据安全传输协议，在不知道密钥的情况下，单从密文信息不能得到原始光照强度数据，若加密的光照数据在不可靠的 WSN 传输过程中发生比特错误或被修改，基站则可通过消息鉴别码快速验证数据的完整性。



(a) 正确解密的光照强度数据



(b) 解密错误时的光照强度数据

图 10　光照强度数据安全解密

4.3　安全性能评估

本文实验选用的 Mica2 节点仅配备 128KB 的 Flash ROM 和 4KB 的 RAM。下面采用 TinyOS 系统的编译器来计算运行本文安全方案的内存耗费，并与 TinyOS 上现有的 TinySec 安全协议进行性能比较。

TinySec 是一套 TinyOS 内置的安全协议，也包括了对数据的加密以及消息认证码的检验。分别测试信息采集节点程序在以下 3 种情况下的存储开销：仅使用 TinyOS 系统的原始通信协议栈、使用 TinySec 安全协议、使用本文的安全方案，性能结果如表 6 所示。

表 6 不同安全方案的 Mica2 节点性能对比

安全方案	ROM/byte	RAM/byte
TinyOS（原始通信协议）	11 858	522
TinySec	20 962	757
本文方案	17 434	654

由于 TinySec 采用了 32 轮运算的 64bit 分组加密算法 SkipJack，而本文的混沌密码算法仅采用 4 轮 8bit Feistel 结构，因此在原始通信协议的基础上，本文的混沌密码运行比 TinySec 更低耗，仅增加了 5 576byte 的 ROM 开销和 132byte 的 RAM 开销，更适用于 WSN 节点。

5 结束语

本文针对无线传感网络节点计算和存储资源受限的特点，首先介绍了一种将 Logistic 混沌映射转化成整数型迭代函数的方法，并在此基础上介绍了一种分组长度为 8bit 的混沌 Feistel 结构，利用 32bit 整数混沌序列迭代值作密钥构建了一个轻量级的 8bit Feistel 分组加密方案；基于整数型混沌映射及 8bit 混沌分组加密算法，设计提出了一种新型的消息鉴别码算法，即利用混沌分组加密和混沌序列构建动态的 S 函数，最终产生输出字节长度可变的 MAC 认证码。大量测试分析表明，本文提出的消息散列鉴别码具有较高的安全性能，大部分测试指标均优于目前一般的混沌散列算法以及 MD5 算法。最后在 Mica2 节点和 TinyOS 平台上，设计了一种基于混沌分组加密和消息鉴别码的数据安全传输应用方案，并给出原型实现与测试结果，且在安全协议数据分组大小与运行内存等方面与 TinySec 协议进行了对比实验，显示了本文方案适用于无线传感网的性能优势。

本文研究未涉及混沌消息鉴别码的初始密钥管理问题，下一步工作将结合作者已提出的一种神经网络密码应用模型[15]，研究 WSN 混沌消息鉴别方案的密钥管理问题。

参考文献：

[1] WATRO R, KONG D, CUTI S. TinyPK:securing sensor networks with public key technology[A]. SASN'04(ACM)[C]. Washington DC, 2004. 322-339.

[2] LIU A, KAMPANAKIS P, PENG N. TinyECC: elliptic curve cryptography for sensor networks (version 0.3)[EB/OL]. http://discovery.csc.ncsu.edu/software/TinyECC, 2007.

[3] 陈铁明, 白素刚,蔡家楣. TinyIBE:面向无线传感器网络的身份公钥加密系统[J].传感技术学报,2009,22(8): 1193-1197.
CHEN T M, BAI S G, CAI J M. TinyIBE: an identity-based encryption system for wireless sensor network[J]. Journal of Transduction Technology, 2009, 22(8):1193-1197.

[4] FAN C, TAN J,ZHENG P. Low-speed wireless networks research and simulation based on RC5[A]. Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing[C]. Beijing, China, 2009. 4521-4524.

[5] The RC6 block cipher[EB/OL]. http://theory.lcs.mit.edu/%7erivest/rc6.pdf, 2006.

[6] Skipjack and KEA algorithm specifications[EB/OL]. http://csrc.nist.gov/CryptoToolkit/skipjack/skipjack.pdf, 2005.

[7] SASTRY N, WAGNER D. Security considerations for IEEE 802.15.4 networks[A]. Proceedings of the 2004 ACM Workshop on Wireless Security[C]. New York, 2004. 32-42.

[8] 陈铁明, 葛亮, 蔡家楣. TinyTCSec:一种新的轻量级无线传感器网络链路加密协议[J]. 传感技术学报, 2011, 24(2):275-282.
CHEN T M, GE L, CAI J M. TinyTCSec: a novel and lightweight data link encryption scheme for wireless sensor networks[J]. Journal of Transduction Technology, 2011,24(2):275-282.

[9] CHEN S, ZHONG X X, WU Z Z. Chaos block cipher for wireless sensor network[J]. Science China Series F-Information Science, 2008, 51(8):1055-1063.

[10] CHEN S, SHU R. BLOCK permutation cipher in chaos with feistel structure for wireless sensor networks[J]. Advances in Intelligent and Soft Computing, 2011, 105:391-296.

[11] WANG Y, LIAO X, XIAO D. One-way hash function construction based on 2D coupled map lattices[J]. Information Sciences, 2008, 178: 1391-1406.

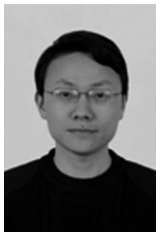
[12] ZHANG H, WANG X, LI Z. One-way hash function construction based on spatiotemporal chaos[J]. Acta Phys Sin, 2005, 54(9):4006-4011.

[13] KANSO A, SMAOUI A. Logistic chaotic maps for binary numbers generations[J]. Chaos, Solitons and Fractals, 2009(40): 2557-2568.

[14] KARLOF C, SASTRY N, WAGER D. TinySec: a link layer security architecture for wireless sensor networks[A]. Proceedings of the 2nd International Conference & Embedded Networked Sensor Systems[C]. Baltimore, MD, USA, 2004. 162-175.

[15] 陈铁明, HUANG H S, 刘多. 神经密码协议模型研究[J]. 计算机研究与发展, 2009, 46(8):1316-1324.
CHEN T M, HUANG H S, LIU D. Neural cryptographic protocol research[J]. Journal of Computer Research and Development, 2009, 46(8):1316-1324.

作者简介：



陈铁明（1978-），男，浙江诸暨人，博士，浙江工业大学副教授，主要研究方向为网络信息安全与智能计算。

葛亮（1986-），男，浙江杭州人，浙江工业大学硕士生，主要研究方向为传感器网络安全协议。