基于分段线性映射与代数运算的混沌密码算法

杨华千, 张 伟, 韦鹏程
(重庆教育学院计算机与现代教育技术系, 重庆 400067)

摘 要: 在本文提出的新的分组密码系统中, 通过迭代一个混沌分段线性映射得到的十进制序列的数字位按照算法 1 构造了一个双射函数 $g(\cdot)$; 通过比较两个混沌分段线性映射产生的十进制序列的对应项得到 $64n$ 比特噪声向量. 经过群上的三种运算(异或运算、模乘运算和模加运算)与由双射函数确定的置换运算交替作用(共 8 轮)在 $64n$ 比特的明文上得到 $64n$ 比特密文. 最后, 从理论和仿真实验两个方面对算法的性能进行了分析.

关键词: 混沌映射; 分组密码; 置换; 群论
中图分类号: TP309. 7 文献标识码: A 文章编号: 0372-2112 (2008) 08-1490-05

A Chaotic Cryptosystem Based on Piecewise Line Map and Algebra Operations

YANG Hua-qian, ZHANG Wei, WEI Peng-cheng
(*Department of Computer and Malern Education Technology, Chongqing Education College, Chongqing 400067, China*)

Abstract: A sequence of decimal numbers generated the chaotic piecewise linear map is used to define a bijection map; two sequences of decimal numbers individually generated by two chaotic piecewise linear maps are used to determine the noise vectors by comparing the element of the two sequences; three operations (bit-by-bit exclusive OR, module multiplication, module addition) in the group and the permutation determined by the bijection map are alternately applied on plaintext with block length 64n bits to produce ciphertext blocks of the same length. Lastly, the performances were analyzed from the theory and experiments.

Key words: chaotic map; block cipher; permutation; group theory

1 引言

大多数的基于混沌的软件加密技术都是使用混沌映射来产生伪随机序列, Kohda 等人在文献[7]中分析了混沌二进制序列的统计特性. Stojanovski 等在文献[5, 6]和 Li Shujun 等在文献[8]中分别提出了用一维分段线性混沌映射(Piecewise Linear Chaotic Map, PLCM)作为密码系统的伪随机数发生器. Xun Yi 和 Jakimoski 等人详细讨论了混沌加密技术在分组密码中的应用及其对密码系统的影响[3,4]. 然而, Wheeler 等人在文[9, 10]中指出当混沌系统用有限精度的计算机来实现的时候, 数字化的混沌系统表现出了许多明显的不同的行为. 它们的数字动力学行为也远不如于连续混沌系统的动力学行为. 例如, 非常短的周期, 依赖于特定的数字精度等[11].

Xun Yi 等人提出了另一种混沌密码系统[3]. 在这个密码系统中, 由混沌 Tent 映射产生的实值序列通过一个域值函数来确定 $4n$ 比特的噪声向量. 同时, 也确定了一个 4 比特位和 1~4 的排列之间的一个查询表. 然后, 噪声向量和排列置换操作交替应用到 $4n$ 比特明文上以产生 $4n$ 比特的密文($n \geqslant 16$). 显然, 该密码系统存在如下两个缺陷: (1)查询表太小, 只有 16 项(因为, 4 比特位至多有 16 种取值); (2) v_{ji} 和 w_{ji} 排列之间的关系是固定不变的, 与密钥无关. 在选择明文攻击下, 这两个缺陷有可能构成密码系统的安全隐患. 在本文中, 一种新的基于混沌和代数群论的更具安全性的混沌分组密码被提出来了. 在这个密码系统中, 通过比较两个混沌分段线性映射产生的十进制序列的对应项得到 $64n$ 比特噪声向量. 同时, 也按照算法 1 定义了一个双射函数 $g: v_{ji} \rightarrow w_{ji}$ 来描述 v_{ji} 和排列 w_{ji} ($1 \sim 8$) 之间的关系.

2 新的分组密码系统

2.1 混沌映射的选择

在文献[8]中提出了一个具有良好随机统计特性

收稿日期: 2007-01-22; 修回日期: 2008-03-10
基金项目: 国家自然科学基金(No. 60573047); 重庆市教委(No. kj061501); 重庆市科委自然科学基金(No. CSTC2006BB2254)

的一维分段线性混沌映射, 其定义如下:

$$F(p,x)=\begin{cases} x/p & ,\ x\in[0,p) \\ (x-p)/(1/2-p) & ,\ x\in[p,1/2] \\ F(p,1-x) & ,\ x\in[1/2,1] \end{cases} \quad (1)$$

此处, p 是控制参数, 且 $p\in(0,1/2)$. 该混沌映射在区间 $[0,1]$ 上具有下面的一些比较好的统计特性[8]:

(1) 其 Lyapunov 指数大于零, 系统是混沌的, 输出信号满足遍历各态性、混和性和确定性.

(2) 具有一致的不变分布密度函数 $f(x)=1$.

(3) 输出轨道的近似自相关函数 $\tau(n)=\delta(n)$.

2.2 新的基于混沌映射和群论的分组加密算法

2.2.1 密钥的选择　本文加密算法的密钥由四个密钥参数 p_1, p_2, x_0, K 构成, 并且要求 $0<p_1, p_2<1/2$, $p_1\neq p_2$, K 是一个 $8\times8=64$ 比特的二进制密钥串. 定义如下两个具有相同初值 (x_0) 的离散分段线性映射来产生拟混沌轨道 $\{x_1(i)\}, \{x_2(i)\}$:

$$F(p_1, x_1(0)): x_1(i+1)=F(p_1, x_1(i)) \quad (2)$$

$$F(p_2, x_2(0)): x_2(i+1)=F(p_2, x_2(i)) \quad (3)$$

此处, $x_1(0)=x_2(0)=x_0$, $i=0,1,2,\cdots$

2.2.2 噪声向量的构造　首先, 分别用离散混沌映射 $F(p_1, x_1(0))$ 和 $F(p_2, x_2(0))$ 产生两个拟混沌序列(为了更好的性能可以让映射先行迭代 N_0 次): $x_1(1), x_1(2), \cdots, x_1(i), x_2(1), x_2(2), \cdots, x_2(i), \cdots$

然后, 定义噪声向量 $\boldsymbol{U}_j=(u_{64j}, u_{64j+1}, u_{64j+63})$, $j=0,1,2,\cdots$, 对任意 u_i[7] 有:

$$u_i=\begin{cases} 0 & ,\text{ if } x_1(i)>x_2(i) \\ \text{no output} & ,\text{ if } x_1(i)=x_2(i) \\ 1 & ,\text{ if } x_1(i)<x_2(i) \end{cases} \quad (4)$$

2.2.3 混淆与扩散过程　对于 $j=0,1,\cdots$, 设 $V_j=(v_{j0}, v_{j1}, \cdots, v_{j7})=(U_{j+2}\oplus K<<<3$. 这里, $\oplus$ 表示按位异或, $<<<3$ 表示循环左移 3 位, $v_{ji}(i=0,1,\cdots,7)$ 是一个 8 位的二进制位串, 即 $v_{ji}\in\{0,1\}^8$.

首先, 构造一个双射函数 $g: v_{ji}\rightarrow w_{ji}$(表 1):

表 1　v_{ji} 和 1, 2, 3, 4, 5, 6, 7, 8 的排列 w_{ji} 之间的映射关系

v_{ji}(8 比特)	w_{ji}(8 比特)	v_{ji}^{-1}(群 $Z_{2^8+1}^*, \odot$ 中)
00000000	w_{j0}	256
00000001	w_{j1}	00000001
……	w_{ji}	……
11111110	$w_{j(2^8-2)}$	10101011
11111111	$w_{j(2^8-1)}$	10000000

即针对每一个 v_{ji} 构造一个 1~8 的排列 w_{ji}. 其构造过程如下: 由混沌映射 $F(p_1, x_1(i))$ 产生一个新的混沌状态 $x_1(i+1)$; 通过模 8 加 1 操作, 抽取 $x_1(i+1)$ 的前 8 个不同的数字位得到 1~8 的一个排列. 如果抽取失败(即, 状态 $x_1(i+1)$ 中的数字位通过模 8 加 1 操作不能得到 1~8 的一个排列)或者得到的排列前面已经出现过, 则继续迭代 $F(p_1, x_1(i))$, 直到得到 256 个不同的 1~8 的排列为止. 实现伪代码如下(算法 1):

算法 **1:**

```
r ←0
  while r ⩽2^8− 1{
    while (1> 0){
      for u ←(1) to (8)
        do P[u] ←0
      x1(i+1) ←F(p1, x1(i))
      x1(i)= x1(i+1)
      u ←1
      while x1(i+1)> 0{
        k= (└x1(i+1)×10┘ mod 8)+1
        if k≠(P[1] to P[u]) then
          P[u] ←k
          u ←u+1
        if u ⩽8 then x1(i+1) ←x1(i+1)×10− floor(x1(i+1)×10)
          else break
      }
      if P[8] ≠0 then break
    }
    if P[1]P[2] …P[8] ≠(wj0 to wjr) then
      wjr ←P[1]P[1] …P[8]
      r ←r+1
}
```

然后, 构造一个置换/代换映射 $f_{ji}(\cdot)$:

设有一个 64 位的二进制位串 $M=(M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8)$, $f_{ji}(\cdot)$ 的定义如下:

$$f_{ji}(M_1, M_2, \cdots, M_k, \cdots, M_8)=[w_{ji}(r_{ji}(M_1), r_{ji}(M_2), \cdots, r_{ji}(M_k), \cdots, r_{ji}(M_8))] \quad (5)$$

其中 $M_k(k=1,2,\cdots,8)$ 是一个 8 位的二进制位串. $r_{ji}(\cdot)$ 表示 M_k 与 v_{ji} 在代数群 $(Z_{2^8+1}^*, \odot)$ 中的模 2^8+1 乘运算, 即

$$r_{ji}(M_k)=M_k\odot v_{ji}=M_k\cdot v_{ji} \bmod (2^8+1) \quad (6)$$

其中 $w_{ji}(\cdot)$ 表示把 $(r_{j1}(M_1), r_{j1}(M_2), \cdots, r_{ji}(M_k), \cdots, r_{ji}(M_8))$ 按照映射 g 中 w_{ji} 所对应的排列进行重新排序. 例如: $v_{ji}=(01100001)_2$, $w_{ji}=(4,6,1,3,5,8,7,2)$, $M_3=(01100010)_2$, 由于 $v_{ji}\odot M_3=(11111110)_2=254$, 则

$$\begin{aligned} &f_{ji}(M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8) \\ &=[w_{ji}(r_{ji}(M_1), r_{ji}(M_2), r_{ji}(M_3), r_{ji}(M_4), \\ &\quad r_{ji}(M_5), r_{ji}(M_6), r_{ji}(M_7), r_{ji}(M_8)] \\ &=w_{ji}[M_1', M_2', (11111110), M_3', M_4', M_5', M_6', M_7', M_8') \\ &=(M_4', M_6', M_1', (11111110), M_5', M_8', M_7', M_2') \\ &=(M_4', M_6', M_1', M_3', M_5', M_8', M_7', M_2') \end{aligned}$$

此处, $M_k'=r_{ji}(M_k)=M_k\odot v_{ji}=M_k\cdot v_{ji} \bmod (2^8+1)$.

当 i 从 0 到 7 时, 记

$$f_j = f_{j7} \circ \cdots \circ f_{ji} \circ \cdots f_{j0} \quad (7)$$

$$f_j^{-1} = f_{j0}^{-1} \circ \cdots \circ f_{ji}^{-1} \circ \cdots \circ f_{j7}^{-1} \quad (8)$$

$$f_{ji}^{-1}(M_1, M_2, \cdots, M_k, \cdots, M_8) = [w_{ji}^{-1}(r_{ji}^{-1}(M_1), r_{ji}^{-1}(M_2), \cdots, r_{ji}^{-1}(M_k), \cdots, r_{ji}^{-1}(M_8))] \quad (9)$$

此处, $r_{ji}^{-1}(M_k) = M_k \odot v_{ji}^{-1} M_k \bullet v_{ji}^{-1} \bmod (2^8+1)$, v_{ji}^{-1} 是 v_{ji} 在群 $(Z_{2^8+1}^*, \odot)$ 中的逆元.

2.2.4 加密/解密过程 将原始的明文 P(二进制位流)按顺序分成 $(P_1, P_2, \cdots, P_r)$ 块. 每块长为 64 比特. 如果最后一块 P_r 不足 64 比特, 则在后面补上 0.

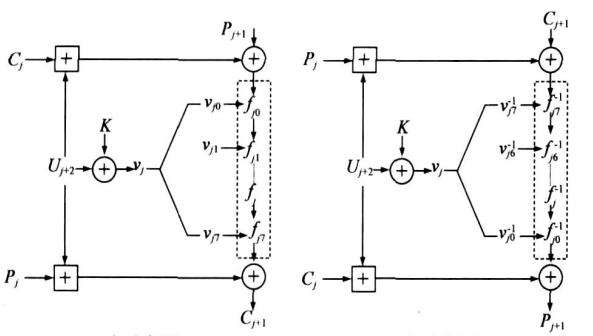
设 $C_0 = U_0$, $P_0 = U_1$. 每一块 64 比特的明文 P_{j+1}, $(j = 0, 1, 2, \cdots, r-1)$ 将按下式(10)加密成 64 比特的密文 C_{j+1}, $(j = 0, 1, 2, \cdots, r-1)$.

$$C_{j+1} = f_j(P_{j+1} \oplus (C_j \boxplus U_{j+2})) \oplus (P_{j+1} \boxplus U_{j+2}) \quad (10)$$

每一个 64 比特的密文 C_j 将按下式(11)解密成 64 比特的明文 P_j

$$P_{j+1} = f_j^{-1}(C_{j+1} \oplus (P_j \boxplus U_{j+2})) \oplus (C_j \boxplus U_{j+2}) \quad (11)$$

其加密/解密结构如图 1. 在整个加/解密过程中, $\oplus$表示在群 $(F_2^{64}, \oplus)$ 中的按位异或, $\odot$ 表示在群 $(Z_{2^8+1}^*, \odot)$ 中的模 2^8+1 乘运算. $\boxplus$表示在群 $(Z_{2^{64}}, \boxplus)$ 中的模 2^{64} 加运算.



(a) 加密框图 (b) 解密框图

图 1 加密与解密框图

3 安全性与性能分析

3.1 密钥空间

在下面的分析中, 采用了 IEEE 754[12]浮点数标准. 设 $p_1 = 0.d_1^{(1)} d_2^{(1)} \cdots d_{15}^{(1)}$, $p_2 = 0.d_1^{(2)} d_2^{(2)} \cdots d_{15}^{(2)}$, $x_0 = 0.x_1 x_2 \cdots x_{15}$. 因为 $0 < p_1, p_2 < 1/2$, $p_1 \neq p_2$, 所以 p_1, p_2 的第 1 位 $d_1^{(1)}$, $d_1^{(2)} \in \{0, 1, 2, 3, 4\}$, 又 K 的长度为 64 位, 则本文提出的算法的密钥空间约为

$$(5 \cdot 10^{14}) \times (5 \cdot 10^{14}) \times (10^{15}) \times 2^{64} = 2.5 \times 10^{44} \times 2^{64} \approx 2^{207.5}$$

如果采取蛮力攻击, 此时密码分析者并不需要知道具体的 p_1, p_2 和 x_0, 但需要知道 U_j, K 和映射 $g: v_{ji} \rightarrow w_{ji}$, 每一个 v_{ji}(8 位, 共有 8! 种取值)所对应的排列 w_{ji} (8 位, 共有 8! 种取值). 此时的密钥空间约为 $256 \cdot 8! \cdot 2^{64} \cdot 2^{64} \approx 2^{151.299}$. 就目前计算能力, 该数字已相当大了.

3.2 扩散与混淆分析

由于群上的三种不同代数运算(群 $(F_2^{64}, \oplus)$ 中的按位异或运算, 群 $(Z_{2^8+1}^*, \odot)$ 中的模 2^8+1 乘运算, 群 $(Z_{2^{64}}, \boxplus)$ 中的模 2^{64} 加运算)任何两种都不满足分配律和结合律, 再加上 $w_{ji}(\cdot)$ 的重排. 所以算法获得了很好的扩散与混淆作用. 下面证明 $(Z_{2^8+1}^*, \odot)$ 是可交换群.

2^8+1 是一个素数, $Z_{2^8+1}^* = \{a \mid a \in 1, 2, \cdots, 256\}$ 对模 2^8+1 乘法运算 $\odot$ ($\odot$, $a \odot b = (a \bullet b)_{2^8+1}$) 构成交换群. 其证明过程如下:

设 $m = 2^8+1 = 257$, 则 m 是一个素数.

(1) $\odot$运算是自封的 假设 $a, b \in Z_m^*$, 即 $0 < a, b < m$. 因为 m 是素数, 所以 $(a, m) = (b, m) = 1$ 且 $(ab, m) = 1$. 设 m 去除 ab 所得的商是 q, 余数是 $(ab)_m$, 即: $ab = qm + (ab)_m$, $0 \leqslant (ab)_m < m$.

所以 $(ab, m) = ((ab)_m, m)$, 所以 $((ab_m, m) = 1$, 所以 $a \odot b = (ab)_m \in Z_m^*$.

(2) $\odot$运算是可交换的 设 $a, b \in Z_m^*$, 则 $a \odot b = (ab)_m = (ba)_m = b \odot a$. 所以, $\odot$运算是可交换的.

(3) $\odot$运算是可结合的 设 $a, b, c \in Z_m^*$. 如果 $m \mid a-b$, 则 $(a)_m = (b)_m$. 由于 $ab - (a)_m (b)_m = a(b - (b)_m) + (a - (a)_m)(b)_m$ 是 m 的倍数. 所以 $m \mid ab - (a)_m (b)_m$. 因此, $(ab)_m = ((a)_m (b)_m)_m$. 另一方面, $c \in Z_m^*$, 则 $c = (c)_m$. 所以

$$\begin{aligned}(a \odot b) \odot c &= ((ab)_m \bullet c)_m = ((ab)_m \bullet (c)_m)_m \\ &= ((ab)c)_m = (a(bc))_m = (a(bc)_m)_m \\ &= a \odot (b \odot c)\end{aligned}$$

(4) 单位元 **e** 的属性 对所有的 $g \in Z_m^*$, 由于 $1 \odot g = (1 \times g)_m = g = (g \times 1)_m = g \odot 1$, 所以 1 是群 $(Z_m^*, \odot)$ 中的单位元.

(5) 逆元的唯一存在性 设 $a \in Z_m^*$, 则 $(a, m) = 1$, 所以存在整数 c, d 使得 $1 = ca + dm$, 所以 $(c, m) = 1$, $((c)_m, m) = 1$, $(c)_m \in Z_m^*$, $1 = ca + dm = (ca + dm)_m = (ca)_m = ((c)_m \bullet a)_m = (c)_m \odot a$. 因此 $(c)_m = a^{-1}$, 逆元的唯一存在性得证.

3.3 加密/解密的唯一性

对于任何一个加密算法, 其加密/解密的唯一性都是必须的. 从第 2 节的描述可以看出, 本文中的算法涉及到以下几种运算: 群上的异或 $\oplus$、模加 $\boxplus$、模乘 $\odot$和根据映射 $g: v_{ji} \rightarrow w_{ji}$ 的重排置换. 因此, 本文提出的加密算法的加密/解密的唯一性是由以下两点保证的:

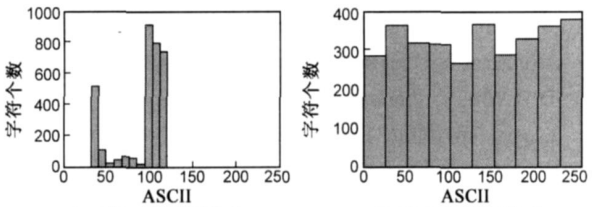
(a) 在群 $(F_2^{64}, \oplus)$, $(Z_{2^8+1}^*, \odot)$ 和 $(Z_{2^{64}}, \boxplus)$ 上的运算是可逆的, 且其逆元是唯一的.

(b) 映射 $g: v_{ji} \rightarrow w_{ji}$ 是一个双射.

3.4 统计测试

通常, 文本文件中的字符都是一些可见字符, 其

ASCII 码位于 033~ 126 之间, 用本文中的算法加密之后, 其 ASCII 码分布于 0~ 255 之间且更加均匀, 因而具有更好的抗统计攻击能力. 为了评估本文提出的算法性能, 约 3200 字节的文本用本文提出的算法进行加/解密, 按照文献[3, 14]的方法进行测试. 实验统计结果(图 2, 表 2) 表明密文的分布完全不同于明文, 其在整个 ASCII 码表上的分布更加均匀.



(a) 明文 ASCII 码分布　(b) 密文 ASCII 码分布

图 2　明文与密文的分布

表 2　相关系数(N= 6400, N= 25600)

	x_0= 0.436567349535648 p_1= 0.485734534345379 p_2= 0.234579834895896	x_0= 0.436567349535647 p_1= 0.485734534345379 p_2= 0.234579834895896	x_0= 0.436567349535648 p_1= 0.485734534345379 p_2= 0.234579834895897
相关系数 (N= 6400)	0.0224	0.0235	0.0242
相关系数 (N= 25600)	0.0118	0.0121	0.0115

3.5　密钥敏感性

在下面的分析与实验过程中, 由于 $0 \notin Z^*_{2^8+1}$ 而 2^8= 256$\in Z^*_{2^8+1}$, 所以用 2^8 代替 0. 另外, 采用扩展的 Euclidean 算法来计算算群($Z^*_{2^8+1}$, $\odot$)的逆元.

设 x_0= 0.436567349535648

p_1= 0.485734534345379

p_2= 0.234579834895896. 密钥 K= "cryption", 其对应的二进制为:

K= 01100011 01110010 01111001 01110000 01110100
01101001 01101111 01101110
c r y p t
i o n

先让 $F(p_1, x_0)$, $F(p_2, x_0)$ 迭代 250 次, 接着按算法 1 构造 g: $v_{ji} \rightarrow w_{ji}$ 的映射(表 3), 最后计算 U_0, U_1, U_2, V_0.

表 3　v_{0i} 和 1, 2, 3, 4, 5, 6, 7, 8 的排列 w_{oi} 之间的映射关系

w_{0i}		w_{0i}		v_{0i}^{-1}
v_{00}	245	54162387	w_{00}	107
v_{01}	163	82457613	w_{01}	41
v_{02}	70	14275386	w_{02}	246
v_{03}	253	31264785	w_{03}	64
v_{04}	209	25417863	w_{04}	91
v_{05}	89	65418732	w_{05}	26
v_{06}	15	74362518	w_{06}	120
v_{07}	168	67385214	w_{07}	231

U_0=
011100001010101100011110001100011010010011000011011010000010000 1

U_1=
000100000001111010010101111101101001010011000001000011100010001 1

U_2=
011111011100011000010001101101111110011100100001001001110100110 11

$V_0 = U_2 \oplus K <<< 3$

= 11110101101000110100011011111101110100 01
v_{00} v_{01} v_{02} v_{03} v_{04}
01011001000011111010100 0
v_{05} v_{06} v_{07}

则明文 P_1= "example 1"(其 ASCII 码: "101, 120, 097, 109, 112, 108, 101, 049") 按照(10)、(5)和(7) 加密后的密文 C_1 的 ASCII 码是"139, 186, 020, 164, 087, 052, 026, 185".

任何一种密码系统一样, 都需提供三种重要特性来防止密码分析[13], 即:

(1) 对密钥敏感: 对同一明文, 密钥的微小变化将产生完全不同的密文; (2) 对明文敏感: 对同一密钥, 明文的微小变化将产生完全不同的密文; (3) 明文到密文的映射是随机的: 一个好的密码系统, 密文中不应该存在任何固定模式.

由于本文算法的密钥是由 p_1, p_2, x_0, K 构成的, 所以将从以下两方面来加以验证:

(1) 保持 p_1, p_2 和 x_0 不变, 改变 K 的最后一位得到 K', 即

K' = 01100011 01110010 01111001 01110000 01110100
01101001 01101111 01101111
c r y p t
i o o

则 P_1= "example 1"加密后的密文 C_1' 的 ASCII 码是"063, 228, 053, 068, 041, 184, 046, 031", 它完全不同于 C_1.

(2) 保持 p_1, p_2 和 K 不变, 改变 x_0 的最后一位得到 x_0'= 0.436567349535647. 则

U_0' =
111001100000100000100011101101000010011011110110010111110000010

U_1' =
000000110110011000001011101110001010100010010110111000000000100

U_2' =
010011011111111100000100110000001000011101110001101001110011011

$V_0' = U_2' \oplus K <<< 3$

= 01110100011010111101101100001111001000
v_{00} v_{01} v_{02} v_{03} v_{04}
11000101111001111010 1001
v_{05} v_{06} v_{07}

明文 P_1= "example 1"加密后的密文 C_1'' 的 ASCII 码是"004, 116, 078, 084, 185, 093, 011, 107".

可以看出, 当密钥 x_0 只有 10^{-15}差异时, 此时参与群中运算的元素及映射 $g(\cdot)$ 与原来的完全发生了变化(见表 3, 表 4). 同时, 得到的密文 C_1'' 也完全不同于 C_1. 另外, 由于本文算法在解密过程中需要 v_{ji} 在群 $(Z_{2^8+1}^*, \odot)$ 中的逆元 v_{ji}^{-1}, 而不同元素的逆元也是不同的, 所以本文的算法对解密密钥也是敏感的.

表 4 v_{0i}' 和 **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8** 的排列 w_{0i}' 之间的映射关系

v_{0i}'		w_{0i}'		$(v_{0i}')^{-1}$
v_{00}'	116	72865341	w_{00}'	113
v_{01}'	107	71548623	w_{01}'	245
v_{02}'	237	75483261	w_{02}'	167
v_{03}'	135	74158623	w_{03}'	99
v_{04}'	152	58316427	w_{04}'	93
v_{05}'	197	42568317	w_{05}'	227
v_{06}'	231	14875632	w_{06}'	168
v_{07}'	169	48132567	w_{07}'	73

4 结论

本文提出了一种基于分段线性混沌映射和群论的分组密码算法. 该算法中的密文依赖于明文、噪声向量、置换运算和代数群上的运算. 它弥补了一些纯混沌密码算法的缺陷. 大的密钥空间、三种群运算的扩散与混淆和排列置换运算保证了新的密码系统对唯密文攻击, 统计攻击及其选择明文攻击等都有很好的抗攻击能力. 我们未来的工作将继续改善算法的性能并在其上进行各种其它的密码分析, 以进一步完善该算法.

参考文献:

[1] Tang G, Liao XF, et al. A novel method for designing S-boxes based on chaotic maps[J]. Chaos, Solitons & Fractals, 2005, 23(2): 413- 419.

[2] Tang G, Liao XF, et al. A method for designing dynamical S-boxes based on discretized chaotic map [J]. Chaos, Solitons & Fractals, 2005, 23(5): 1901- 1909.

[3] Xun Yi, Chik How Tan, and Chee Kheong Siew. A New Block Cipher Based on Chaotic Tent Maps [J]. IEEE Trans. Circuits and Systems I, 2002, 49(12): 1826- 1829.

[4] G Jakimoski, L Kocarev, et al. Chaos and cryptography: Block encryption ciphers based on chaotic maps[J]. IEEE Trans. Circuits and Systems I, 2001, 48(2): 163- 169.

[5] T Stojanovski, L Kocarev, et al. Chaos-based random number generators—Part I: Analysis[J]. IEEE Trans. Circuits and Systems I, 2001, 48(3): 281- 288.

[6] T Stojanovski, L Kocarev, et al. Chaos-based random number generators—Part II: Practical realization[J]. IEEE Trans Circuits and Systems I, 2001, 48(3): 382- 385.

[7] Tohru Kohda, Akio Tsuneda. Statistics of Chaotic Binary Sequences[J]. IEEE TRANSACTIONS ON INFORMATION THEORY, 1997, 43(1): 104- 112.

[8] Li Shujun, Mou Xuanqin and Cai Yuanlong. Pseudo-Random Bit Generator Based on Couple Chaotic Systems and its Applications in Streamr-Cipher Cryptography[A]. Progress in Cryptology-IndoCrypt 2001[C]. London, UK: Springer-Verlag, Lecture Notes in Computer Science, 2247: 316- 329.

[9] D D Wheeler. Problems with chaotic cryptosystems [J]. Cryptologia, 1989, XIII(3): 243- 250.

[10] D D Wheeler, R A J Mathews. Supercomputer investigations of a chaotic encryption algorithm [J]. Cryptologia, 1991, XV(2): 140- 152.

[11] Jun Wei, Xiaofeng Liao, Kwok-wo Wong and Tao Xiang. A new chaotic cryptosystem[J]. Chaos, Solitons and Fractals, 2006, 30(5): 1143- 1152.

[12] D Goldberg, D Priest. What every computer scientist should know about floating-point arithmetic[J]. ACM Comp. Surv., 1991, 23(1): 5- 48.

[13] J Fridrich. Image encryption based on chaotic maps [A], Systems, Man, and Cybernetics, 1997. apos; Computational Cybernetics and Simulationapos[C]. Orlando, FL, USA: 1997 IEEE International Conference on Volume 2. 1105- 1110.

[14] D E Knuth. The Art of Computer Programming [M]. 3thed. Volume 2(Seminumerical algorithms): Addison Wesley, 1998.

作者简介:

杨华千 男, 1972 年 2 月出生, 博士, 研究方向是信息安全. 主持或参研国家自然科学基金、重庆市科委自然科学基金和重庆市教委基金项目 10 余项. 获重庆市科技进步三等奖 1 项, 发表学术论文 40 余篇. 其中: SCI 收录 10 篇、EI 收录 5 篇、ISTP 收录 4 篇. 建设国家精品课程 1 门, 主编普通高等教育"十一五"国家级规划教材 1 部. E-Mail: mailtostorm@163.com

张 伟 男, 1970 年 6 月出生, 教授, 博士后, 留澳访问学者. 研究方向是信息安全、计算智能与数据挖掘. 主持或参研国家自然科学基金项目等 22 项科研课题, 获重庆市自然科学二等奖 1 项、重庆市科技进步三等奖 1 项, 发表学术论文 100 余篇. 其中: SCI 收录 13 篇、EI 收录 17 篇、ISTP 收录 6 篇, 第一作者 SCI 论文被他引 11 次. 建设国家精品课程 1 门, 获重庆市高等教育教学成果二等奖 1 项, 主编普通高等教育"十一五"国家级规划教材 2 部, 培养指导的学生在全国、省(市)竞赛中 16 次获奖.