

基于离散斜帐篷映射的混沌加密系统

王颖学

(重庆大学 计算机学院, 重庆 400044)

(happybear3000@126.com)

摘要:利用帐篷映射的混沌特性并结合动态参数和明文块细分子块方法,提出了一种改进了的基于离散斜帐篷映射的混沌加密方法。该方法不仅继承了原有系统的优良密码学特性,而且通过理论分析和实验证明它拥有更好的抗选择明文攻击特性以及较稳定的加解密速度。

关键词:帐篷映射;混沌;选择明文攻击

中图分类号: TP309.7 **文献标识码:** A

A new chaotic cryptosystem based on discrete skew tent map

WANG Ying-xue

(College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: A new chaotic cryptosystem based on discrete skew tent map was proposed by making use of the dynamic encryption parameter and fractionized sub-blocks of plain text block scheme, and taking into account of the chaos features of the tent map. The proposed cryptosystem inherited the advantages of the original one. Theoretic analysis and experiment results prove that it has better anti-chosen-plaintext-attack property and stable encryption/decryption speed.

Key words: tent map; chaos; chosen plain text attack

虽然已有许多新的混沌分块加密系统被提出来,但是这些加密系统的安全性都不够高,线性和差分分析易于攻击这些系统。同时在密码学中,加密速度的快慢对其实现是极其重要的。在文献[1]中,为了设计出快速且能够抵抗差分分析和线性分析的混沌加密系统,Masuda 和 Aihara 提出了一种基于离散帐篷映射的加密系统。本文在该方案基础上改进了该加密系统,并对加密系统的安全性做了相应分析。理论分析和实验表明本文提出的改进算法不仅继承了原方案的优良特性,且有较好的抗选择明文攻击特性。

1 基于有限状态的帐篷映射的加密系统

Masuda 和 Aihara 提出了一种基于一维离散斜帐篷映射的加密系统。其斜帐篷映射的离散化方法比较新颖,易于推广到一般的混沌映射上。该算法将本是二对一的斜帐篷映射通过离散化变为一一对应的离散映射;同时保留了原帐篷映射的混沌特性,如对初值的敏感性和迭代轨道序列的相关性以指数递减。其轨道序列可以视为贝努利序列,即具有强的伪随机特性,可以直接用于对信息的加密,同时因为逆变换的存在,其解密算法将变得非常简单。

1.1 离散化混沌映射为一一对应的离散变换的方法

斜帐篷映射是一种推广的帐篷映射,其定义如下:

$$f_a(x) = \begin{cases} \frac{x}{a}, & 0 < x \leq a \\ \frac{x-1}{a-1}, & a < x \leq 1 \end{cases} \quad (1)$$

该函数是二对一的映射,没有一一对应的逆函数,但可以看作具有两个分支的逆函数,即 $f_a^{-1} = ax$ 或 $f_a^{-1} = 1 + (a-1)x$ 。若想直接利用 $f_a(x)$ 的迭代来加密明文,然后用其逆函数直接解密,则其逆函数的不唯一将使解密面临一致性问题,

解密不能恢复出原文。为了做到解密的一致性,文献[1]提出了如下的离散化方法,以诱导出一个有限状态的一一对应的扩散变换。

令 $M > 2$, 定义如下两个集合, $P = C \equiv \{x; x = 1/M, 2/M, \dots, M/M\}$, 由(1)式诱导的变换关系定义如下:

$$\tilde{f}_a(x) = \frac{|\{x' \in P \mid f_a(x') < f_a(x)\}| + 1}{M} \quad (2)$$

这里 $|\cdot|$ 表示集合的势。这样(2)式就是一个有限状态的离散映射。 $f_a(x)$ 对应到 $\{x; x = 1/M, 2/M, \dots, M/M\}$ 中的第 $\tilde{f}_a(x)$ M 个最小元素,若 $f_a(x_1) = f_a(x_2)$ 且 $x_1 < x_2$, 则令:

$$\tilde{f}_a(x_1) + 1/M = \tilde{f}_a(x_2) \quad (3)$$

容易看出, $\tilde{f}_a(x)$ 是一一对应的函数。并保持了将 $0 < x \leq a$ 和 $a < x \leq 1$ 都拉伸到 $0 < x \leq 1$ 上。其逆变换 \tilde{f}_a^{-1} 如下,若 $\tilde{f}_a(x')$ 是 $\{x; x = 1/M, 2/M, \dots, M/M\}$ 中 $M \cdot y$ 的一个最小元,则令 $\tilde{f}_a^{-1}(y) = x' \in P$ 即可。

1.2 基于离散化斜帐篷映射的加密算法

将(2)式改写为下面的形式:

$$\tilde{f}_a(x) = \begin{cases} \frac{1}{M} \lceil \frac{x}{a} M \rceil, & 0 < x \leq a \\ \frac{1}{M} \lfloor \frac{x-1}{a-1} M \rfloor + M^{-1}, & a < x \leq 1 \end{cases} \quad (4)$$

而 $\tilde{f}_a^{-1}(x)$ 可写为如下的形式:

$$\tilde{f}_a^{-1}(x) = \begin{cases} x_1, & m(y) = yM + 1, \frac{x_1}{a} > \frac{1-x_2}{1-a} \\ x_2, & m(y) = yM, \frac{x_1}{a} \leq \frac{1-x_2}{1-a} \end{cases} \quad (5)$$

为了将(4)式和(5)式用于加密和解密算法,需要将它们进一步修改为整数集合上的变换和逆变换。变换的定义域和值域做如下修改:

令 $X = Mx, Y = My, A = Ma, P' = C' = \{X; X = 1, 2, \dots, M\}, K' = \{A; A = 1, 2, \dots, M\}$ 。

其中 P', C', A' 可分别视为明文空间、密文空间和密钥空间。加密算法就是直接的对明文做 n 次函数迭代得到密文,每一轮都采用如下的加密变换:

$$\tilde{F}_A(x) = \begin{cases} \lceil \frac{M}{A} X \rceil, & 0 < X \leq A \\ \lfloor \frac{M}{M-A} (M-x) \rfloor + 1, & A < x \leq M \end{cases} \quad (6)$$

解密过程是加密的逆过程,对密文做如下的逆变换 n 次即可恢复明文。

$$\tilde{F}_A^{-1}(x) = \begin{cases} X_1, m(Y) = Y + 1, \frac{X_1}{A} > \frac{M - X_2}{M - A} \\ X_2, m(Y) = Y, \frac{X_1}{A} \leq \frac{M - X_2}{M - A} \end{cases} \quad (7)$$

这里, $X_1 = \lfloor M^{-1}AY \rfloor, X_2 = \lceil (M^{-1}A - 1)Y + M \rceil, m(Y) = Y + \lfloor \frac{AY}{M} \rfloor + 1$ 。

2 改进的离散斜帐篷映射加密算法

离散斜帐篷映射的迭代轨道有很好的伪随机特性。不论是从欧氏空间来看,还是从 $GF(2^n)$ 来看,随着迭代次数的增加,在密文中的明文信息以指数速率迅速衰减。这个特性若能很好地利用可以设计出高安全的加密系统。这里,我们将提出一种具有动态参数的离散斜帐篷映射的加密系统。

在文献[1]的加密系统中,密钥是参数 A ,且在加密过程中, A 是没有变化的,这也保证了解密的一致性。而在传统的加密方案中,如高级加密标准(Advanced Encryption Standard, AES)^[2],其密钥在进行时有一个密钥扩散过程,也称为密钥编排。即加密和解密算法使用从种子密钥的字节数组生成的密钥次序表,实质上,从初始密钥生成多个密钥(而非使用单个密钥)会大大增加位的扩散,在对一个明文块加密时,每一轮的加密密钥都是由种子密钥通过扩散过程产生的轮密钥。测试和分析表明对明文块加密采用轮密钥可以加强系统的安全性。

我们将采用动态改变加密参数 A 的方法来达到类似于 AES 的密钥编排方案,以加强系统的安全性。

2.1 基于动态参数的离散斜帐篷映射加密算法

动态改变加密参数 A 是指,在对明文迭代加密时,每次迭代的离散斜帐篷映射的系统参数是不同的。因此,需要有一个在每一轮由种子密钥产生每轮迭代所用的参数的算法。在下面的算法描述中,把这些参数称为轮密钥或子密钥。设种子密钥为 K ,每一轮迭代所需的子密钥记为 $A_i, i = 1, 2, \dots, N$ 。则产生 A_i 的算法如下:

$$A_i = F_{a_i}^{n_i}(i) \quad (8)$$

这里,第 i 轮的子密钥 A_i 是通过(6)式迭代得到的,其所需的迭代次数 n_i 和系统参数 a_i 由种子密钥 K 产生。具体产生 n_i 和 a_i 的算法如下:设明文空间、密文空间和密钥空间的大小按比特长度计为 $L = \lceil \log_2 M \rceil$ 。将种子密钥串划分为基本相等的 q 份,前 $q-1$ 个长为 $\text{floor}(L/q)$,最后一个长为 $L - (q-1) \times \text{floor}(L/q)$,这里 $\text{floor}(\cdot)$ 表示地板函数。即将种子密钥

分为 q 个子串,分别记为 K_1, K_2, \dots, K_q 。下面我们利用 $K_j, j = 1, 2, \dots, q$ 来产生轮密钥,用如下算法产生 n_i 和 a_i :

$$a_i = \begin{cases} (K_i)^i \bmod (M-2) + 1, & 1 \leq i \leq q \\ (K_{1+(i \bmod q)})^i \bmod (M-2) + 1, & q < i \leq N \end{cases}$$

$$n_i = \begin{cases} i \times K_i \bmod 2^8 + \text{floor}(2.39 \times L), & 1 \leq i \leq q \\ i \times K_{1+(i \bmod q)} \bmod 2^8 + \text{floor}(2.39 \times L), & q < i \leq N \end{cases} \quad (9)$$

下面用一个实际的例子来说明上述的密钥编排方案,这里用较短的密钥来演示我们提出的加密方案的有效性。取密钥空间大小为 $M = 2^{16}$,则 $L = 16$,令密钥 $K = 55147 = 1101011101101011$,取 $q = 2$,将密钥分为 2 个子串 $K_1 = 11010111 = 215$ 和 $K_2 = 01101011 = 107$ 。加密迭代的次数 $N = \lceil 2.39 \times 16 \rceil + 15 = 54$ 。

依据(8)式和(9)式,利用 K_1 和 K_2 来产生的 54 轮密钥如下:

$$A_1 = F_{a_1}^{n_1}(1) = F_{216}^{253}(1) = 46057$$

$$A_2 = F_{a_2}^{n_2}(2) = F_{11450}^{252}(2) = 39710$$

\vdots

$$A_{54} = F_{a_{54}}^{n_{54}}(54) = F_{216}^{253}(54) = 13632$$

表 1 $L = 16$, 密钥 $K = 55147 = 1101011101101011$ 产生的各轮子密钥

轮数	轮密钥	轮数	轮密钥	轮数	轮密钥	轮数	轮密钥
1	46057	15	53621	29	13181	43	3733
2	39710	16	53850	30	45066	44	54593
3	46304	17	8065	31	5565	45	4679
4	57248	18	58476	32	29895	46	19609
5	28195	19	17766	33	4528	47	61546
6	34252	20	2674	34	2166	48	23130
7	48080	21	49585	35	14401	49	51572
8	45449	22	43932	36	21717	50	5410
9	12523	23	63283	37	7712	51	1819
10	59425	24	1638	38	25554	52	13798
11	5899	25	37611	39	63724	53	46916
12	26703	26	59491	40	35321	54	13632
13	57288	27	34707	41	5071		
14	29904	28	48938	42	42362		

采用文中提出的加密算法对 256×256 的灰度图像 moon surface 加密和解密的效果见图 1,加密所用的时间为 0.40s,解密所用的时间为 0.53s,即加密速度和解密速度分别为 1.280Mb/s 和 0.96Mb/s (Matlab 实现,并在拥有 2.1GHz 的 CPU 和 512M 内存的个人电脑上运行测试)。

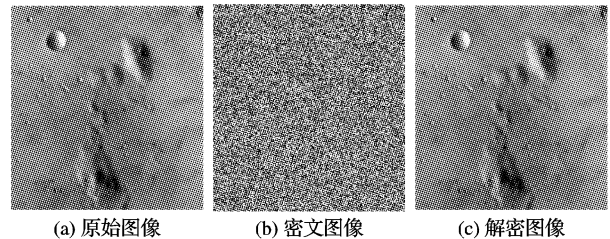


图 1 moon surface 的加密和解密效果

2.2 基于动态参数和明文块细分子块的加密算法

当需要很高的安全程度时,在文献[1]的加密系统中,对应的加密明文和密文空间也会变得很大,这时其上的变换运算的速度将变得很慢,也就是随着密钥长度的增加,加密和解密所需的时间也会变得较长。在许多实时应用中,这是不希望出现的。而在我们改进的加密算法中,由于应用了轮加密

参数动态产生的策略,若对加密和解密算法做一些改进,就可以做到在加密变换的空间大小不变的情形下,扩大明文、密文和密钥的空间。即在要求的加密密钥长度很长时,通过密钥编排,将密钥信息逐渐转移到较短的轮密钥中,即每次迭代变换的参数 A_i 中,以实现在扩展密钥长度时,加密和解密运算的速度基本不发生改变或者增加的计算量很小。

进一步改进的加密算法可视为基于密钥编排和明文块细分子块加密的方案。块细分子块加密的方案的加密和解密算法描述如下:

加密算法 设明文块 P 的长度为 L ,在做加密变换前将明文块分为 $2s$ 个等长的小块 P_1, P_2, \dots, P_{2s} ,子密钥长度为 L/s ,记为 A_1, A_2, \dots, A_s ,它们由(8)式和(9)式根据种子密钥 K 来产生。加密变换的空间大小为 $2L/s$,即每次变换在两个小块上进行。首先将块 P_1 和 P_2 连成一个长为 $2L/s$ 的块,基于变换公式(6),利用子密钥 A_1 对 P_1P_2 做变换得到 $C_1^1C_2^1$,将 $C_1^1P_3$ 连接为一个块,利用子密钥 A_2 对 $C_1^1P_3$ 做变换,得到 $C_2^2C_3^1$;然后再将 $C_3^1P_4$ 连接为一个块,用子密钥 A_3 对 $C_3^1P_4$ 做变换,得到 $C_3^2C_4^1$;对所有 P_1, P_2, \dots, P_{2s} 进行上述的子块加密,得到 $C_{2s-1}^2C_{2s}^1$ 后,连接 $C_{2s-1}^2C_{2s}^1$,利用子密钥 A_{2s} 对 $C_{2s-1}^2C_{2s}^1$ 做变换,得到 $C_{2s}^2C_1^1$ 。至此,我们可以将中间密文块 $C_1^2, C_2^2, \dots, C_{2s}^2$ 视为经过两轮变换后的密文块,且每次变换所用的子密钥是不同的。利用子密钥组 $A_{2ixs+1}, A_{2ixs+2}, \dots, A_{2ixs+s}, i = 1, 2, \dots, N-1$,可以对明文子块再做 $2(N-1)$ 轮变换,得到最终的密文块 $C_1^{2N}, C_2^{2N}, \dots, C_{2s}^{2N}$ 。

解密算法 是加密的逆过程。对密文块 $C_1^{2N}, C_2^{2N}, \dots, C_{2s}^{2N}$,采用子密钥组 $A_{2(N-1)xs+1}, A_{2(N-1)xs+2}, \dots, A_{2Nxs}$,基于逆变换公式(7),经过类似加密过程对细分的明文子块做逆变换,可以得到中间密文块 $C_1^{2(N-1)}, C_2^{2(N-1)}, \dots, C_{2s}^{2(N-1)}$;继续对 $C_1^{2(N-1)}, C_2^{2(N-1)}, \dots, C_{2s}^{2(N-1)}$ 做逆变换,经过 N 次这样的操作,就可以恢复明文块。

采用上面的加密方案就能在较小的变换空间里获得很强的加密安全性,密钥 K 的长度可以远远超出每轮加密的变换空间的尺度 M 。下面用一系列的试验来验证该方案的有效性和进行加密速度的测试。表 2 给出了 3 幅大小不同图像 Moon surface, Lenatestpattern 和 Man 加密和解密的时间与密钥长度的关系。(Matlab 实现,并在拥有 2.1GHz 的 CPU 和 512M 内存的个人电脑上运行测试)。

表 2 加密和解密时间(单位:s)与密码长度关系

操作	对象	大小 /MB	密钥长度/bit			
			64	128	512	1024
加密	Moon surface	0.5	0.98	0.90	1.09	2.31
	Lenatestpattern	2	2.30	2.60	2.50	3.80
	Man	8	9.00	10.40	10.60	11.20
解密	Moon surface	0.5	0.90	0.90	0.81	0.82
	Lenatestpattern	2	3.40	2.90	3.30	3.50
	Man	8	9.50	11.50	12.40	13.10

3 加密算法的安全分析

3.1 抗选择明文攻击

考虑加密系统(6),对任何密钥 $A \in K'$,明文 M 总是被映射到 1。这个特性是该加密系统的一个根本弱点。基于此弱点文献[3]提出如下的攻击方案。

当密文 $C_M = \tilde{F}_A^{(N)}(M)$ 和 $C_1 = \tilde{F}_A^{(N)}(1)$ 已知时,下面的等式总是成立的:

$$\tilde{F}_A^{(N+1)}(M) = \tilde{F}_A^{(N)}(1) = \tilde{F}_A^{(1)}(C_M) = C_1 \quad (10)$$

但我们不知道 $C_M \ll A$ 还是 $C_M > A$,但密钥的可能空间能够表示如下:

$$\left[\frac{M}{A} C_M \right] = C_1 \quad (11)$$

由(10)式和(11)式得到的密钥空间可能超出密钥的范围 K' ,因此在下面的分析中只考虑如下可能的密钥空间:

$$\left(\left[\frac{MC_M}{C_1}, \frac{MC_M}{C_1 - 1} \right] \cup \left[\frac{M(C_1 - 1) - M(M - C_M)}{C_1 - 1}, \frac{MC_1 - M(M - C_M)}{C_1} \right] \right) \cap K' \quad (12)$$

基于一对选择明文的攻击方案可描述如下:若明文对 1 和 M 的密文已知,密钥的可能空间可由(12)式得到,若该集合的势很小,则我们可以通过强力搜索得到真实的密钥。

而在我们的改进方案中,由于都引进了密钥编排,在改进的加密方案中(10)式不再成立,那么相应的(12)式也无法找出,则此类明文攻击将不再有效。

3.2 混乱和扩散特性分析

文献[4]中提出了密码学设计的两个重要原则:混乱原则和扩散原则。

混乱原则 人们所设计的密码应使得密钥和明文以及密文之间的依赖关系相当复杂,以至于这种依赖性对密码分析者来说是无法利用的。

扩散原则 人们所设计的密码应使得密钥的每一位数字影响密文的许多位数字以防止对密钥进行逐段破译,而且明文的每一位数字也应影响密文的许多位数字以便隐蔽明文数字的统计特性。

在我们提出的方法中用一个类似于数据加密标准(Data Encryption Standard, DES)加密算法中 S 盒的方法(混沌变换及其逆变换)来达到混乱的要求,同时又继承了文献[1]中加密算法相同的明文和密钥敏感性。

4 结语

在改进的加密方案中,采用混沌变换(6)式和逆变换(7)式作为加密和解密的基本变换,它类似于 DES 加密算法中 S 盒的作用:对明文进行混淆。改进的加密方案继承了文献[1]中加密算法相同的明文和密钥敏感性,使该加密系统具有良好的抗差分特性和统计特性,被加密的明文信息随着加密轮数呈指数衰减,明文比特和密文比特具有良好的独立性等密码学特性。同时改进方案具有更好的抗选择明文攻击的能力和稳定的加解密速度。

参考文献:

- [1] MASUDA N, AIHARA K. Cryptosystems with discretized chaotic maps[J]. IEEE Transactions on Circuits and Systems, 2002, 49(1): 28-40.
- [2] SCHNEIER B. Applied cryptography-protocols, algorithms, and source code in C[M]. Second Ed. New York, John Wiley & Sons, 1996.
- [3] CHEN Y, LIAO X, WONG K-W. Chosen Plaintext attack on a cryptosystem with discretized skew tent map[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2006, 53(7): 527-529.
- [4] SHANNON CE. Communication Theory of Secrecy Systems[J]. Bell System Technology Journal, 1949, 28(4): 656-715.