基于整数 logistic map 的伪随机数产生法和多维坐标法

董际国[1] 宋寒松[2]
1. 上海理工大学光电信息与计算机工程学院
2. 上海海潮新技术研究所

摘 要：首先，我们介绍一个利用整数来计算混沌函数 logistic map，并对由计算得来的数值系列加以搅拌来产生均匀分布的伪随机数的方法。其次，我们介绍一个用于管理大量的比较短的伪随机数（密钥，认证码）的多维坐标法。利用上述伪随机数产生法和多维坐标法，我们可以容易地实现对大量（比如：2^{64}）的密钥和认证码的管理，可以容易地实现被认为是原理上安全的一钥一密。

关键词：整数 logistic map，多维坐标法，一钥一密

1 前言

一般，随机数（包括密钥，认证码等）的产生和管理是属于不同的研究范畴。安全的随机数的高速产生是由非线性的伪随机数产生器来实现。而随机数的安全管理主要是由对保存在记忆媒体中的随机数进行加密或者加以访问控制来实现。

此前，我们面临着这样的问题。即使有一个好的伪随机数产生器，也无法实现高速安全的流加密，因为流加密必须是一钥一密的。同样，也无法实现一码一认证的安全的认证方式。实现一钥一密的流加密和一码一认证的安全的认证方式所需要的大量的伪随机数的安全可行的管理方法的问题一直困扰着我们。

我们的目的不仅是提供一个好的伪随机数产生方法，还包括，可实现一钥一密所需要的大量的伪随机数的安全可行的管理方法。

本文中，我们首先介绍利用混沌函数 logistic map 来产生伪随机数的方法 [1]。我们用整数来计算混沌函数 logistic map。对所产生的疑似混沌数值系列进行适当的搅拌，对搅拌所产生的数值系列进行了统计检定，确认了搅拌所产生的数值系列具有统计学上的均匀分布特性。我们确认了在有限精度（计算机）计算下的 logistic map 的非线性性和初始值敏感性。但是，在本文中，由于篇幅的限制，我们省略了有关在有限精度计算下的 logistic map 的一些重要的性质的介绍 [2]。

其次，我们介绍一个用来产生和管理大量的比较短的伪随机数的多维坐标法。混沌系列的一个主要特征，就是由相对简单的规则所支配的不规则震动。它提示我们，可以设计一个利用规则来管理无规则（随机数）的系统。多维坐标法就是通过管理有规则的多维坐标来实现对大量（比如：2^{64}）的无规则的伪随机数（密钥，认证码）——key 的管理。

利用上述两方法，我们可以容易地构筑可以产生和安全地管理大量的 key（密钥，认证码）的系统。由此，我们可以容易地实现在原理上安全的一钥一密的加密方法和一码一认的认证方法。

本文的构成如下。2. 是有关利用整数来计算 logistic map 并产生具有良好的统计学特性的伪随机数的论述，3. 是有关可用来产生和管理大量的 key 的多维坐标法。最后，4. 是结言。

2 基于整数 logistic map 的伪随机数产生法

简单的函数可以产生复杂的数值系列，混沌函数被期待用来产生随机数。但是，用

计算混沌函数来产生随机数，主要存在着以下的几个问题：

数值系列的非均等分布性

一般，混沌函数所产生的数值系列并不具有均等分布性。如 logistic map 所产生的数值系列具有 U 字型的分布特性。

逐次决定性

对混沌函数所产生的数值系列…, x_{t-1},x_t ,x_{t+1},…，从 x_t 可以计算 x_{t+1}。因此，混沌函数所产生的数值系列需经变换才能成为安全的随机数。

用浮点计算法计算

一般，混沌函数的定义域为实数。在计算机上，无理数是用浮动小数点计算法来进行近似计算，而浮点计算法存在着不同的规格。因此，在计算混沌函数时，即使输入相同的初始值，也有可能产生出不同的数值系列。而要产生具有长周期性的混沌数值系列，则需要扩张计算精度。而较为复杂的浮点计算法的计算精度的扩张会使计算速度更为低下。另一方面，价格较低的产业用微型计算机大多不支持浮动小数点计算法。因此，计算混沌函数的系统成为相对高价的系统。

2.1 混沌函数 logistic map(LMAP)

早在 1947 年，现代计算机的发明者 Von Neumann 就注意到了 LMAP

$$x_{t+1} = 4x_t(1-x_t), \quad 0 < x_t < 1, \quad t = 0,1,... \tag{2.1}$$

可以产生复杂的数值系列。但其产生的 xt 的系列具有 U 字型分布，而不是均等分布。

通过考察，我们发现式 (2.1) 可以用固定小数点法来计算。而区间（0,1）的固定小数点法的计算，在计算机上可以直接用整数来进行。

2.2 整数 LMAP

用整数来对式 (2.1) 进行固定小数点计算，我们称之为整数 LMAP：

$$X_{t+1} = \lfloor 4X_t(2^N - X_t)/2^N \rfloor \tag{2.2}$$

$1 \le X_t < 2^N, t = 0,1,...$，$\lfloor\ \rfloor$ 为舍去小数点以下的部分的计算。

对于式 (2.2)，我们可以通过对 X_t 进行适当的分割，并有效利用计算机所备的乘法器来实现高速计算。但是，X_t 的系列和 x_t 同样具有 U 字型分布，而不是均等分布。

我们注意到在用式 (2.2) 对 N 比特的 X_t 的计算过程中，存在着 2N 比特的内部状态，并提出了利用这 2N 比特的内部状态来产生 N 比特的随机数的方法。

2.3 随机数的产生：搅拌

我们将搅拌后所得到的数值记为 R_t，则搅拌可以下式来表示：

$$R_t = \lfloor 4X_t(2^N - X_t)/2^N \rfloor \oplus (4X_t(2^N - X_t)\%2^N) \tag{2.3}$$

式 (2.3) 的搅拌方法只用一次 XOR 的操作，能够产生与计算精度相同的 N 比特的随机数，可以说是最有效率的搅拌方法。式 (2.3) 中的符号 为求余的计算。

2.4 效果

2.4.1 统计检定

我们对用上述搅拌方法产生的随机数，进行了如下 8 项的统计检定。

1. equidistribution test
2. serial test
3. poker test
4. coupon collector’s test
5. run test
6. collision test
7. birthday spacings test
8. serial correlation test

在这里，我们省略统计检定详细内容的介绍。其结果表明上述搅拌方法所产生的伪随机数，具有良好的统计学的均等分布特性 [1]。

2.4.2 随机数产生速度

随机数的产生速度依存于所使用的计算机和编程方法。表 1 为各计算精度时的随机数的产生速度的一例。表中，N 为计算精度，S 为随机数的产生速度，PC 为使用 Genuine Intel(R) 1.50GHz 的笔记本电脑，MC 为使

用 ATmega168V 8MHz 的 8 位微型计算机。

表 1. 计算精度和随机数产生速度

N(bit)		32	64	96	128	160	192	224	256
S	PC(Gbps)	1.9	1.0	0.71	0.56	0.51	0.43	0.36	0.31
	MC(Kbps)	262	166	125	101	80	67	60	54

2.5 小结

本伪随机数产生方法用整数的乘法，加法，移位，反转来对混沌函数进行计算，从而以小的计算量实现了所产生的随机数具有非线性性。而其特殊的搅拌方法则以最少的计算量实现了所产生的伪随机数具有均匀分布的特性。由于混沌函数的计算具有逆向不可算性（一方向性），又由于搅拌方法采用了 XOR 计算，而 XOR 计算同样具有逆向不可算性，从而使得所产生的伪随机数具有不可推测性。也就是，在计算精度足够长的条件下，所产生的伪随机数是可应用于信息安全领域的，安全的伪随机数。

几乎所有的计算机（包括产业用微型计，单片机）都装备有整数的乘法器，因此，我们可以很容易地实现费用和性能的最佳组合，以较低的费用取得较高的性能。

3 多维坐标法

利用电子系统的认证技术已经在很多的领域被使用。而经由公共空间的认证却容易被监听。防止对认证码的监听，一码一认证（一次性 key）是有效的。但是，对所使用的认证码必须安全地保存和管理好。

在加密方面，现在主流的是分组加密。原因在于分组加密的密钥可以多次使用。而众所周知，密钥的使用期越长，密码被解读的可能性就越大。也就是说，一钥一密（一次性 key）是最安全的密码。但是，同样存在着密钥的安全保存和管理的问题。

对于 key 的管理，一般，key 或其产生的有关信息等作为秘密信息，保存于储存媒体中，通过加密或访问控制来保护。这样的 key 的管理方法，在具有大量的用户的系统中使用一码一密的加密或者一码一认证时，需要巨大的保存和管理费用。

本文介绍一个，把 key 的产生和管理进行统一处理的，key 的管理方法。本手法称为多维坐标法。

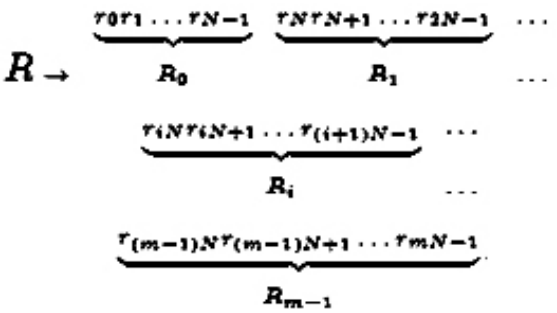

图 1. $G:\{0,1\}^N \to \{0,1\}^{mN}(m:M,r_i:\{0,1\})$

3.1 多维坐标法

对于自然数 $N \geq 1, m \geq 2$，任意的 CSB 产生器（cryptographically strong pseudorandom bit generators）$G: I_N \to I_{mN}$ 作为伪随机数产生器，m 个内部函数 $G_i: I_N \to I_N (i = 0,\ldots,m-1)$ 定义如下。首先，从任意的种子 $R \in I_N$ 产生 mN 比特的 2 值系列 $G(R) = r_0 \cdots r_{mN-1}$（图 1）。

其次，$G_0(R)$ 作为 $G(R)$ 的最先的 N 比特（就是 $G_0(R) = r_0 \cdots r_{N-1}$，$G_1(R)$ 作为 $G(R)$ 的其次的 N 比特（即 $G_1(R) = r_N \cdots r_{2N-1}$），…，$G_{m-1}(R)$ 作为 $G(R)$ 的最后的 N 比特（就是 $G_{m-1}(R) = r_{(m-1)N} \cdots r_{mN-1}$），一般，使 $G_j(R) = r_{jN} \cdots r_{(j+1)N-1}$，$0 \leq j \leq m-1$ 进而，$i = (i_1,\ldots,i_d,\ldots,i_D)$ 作为 D 维坐标时，（ $i_d \in \mathrm{M_d} = \{0,\ldots,M_d - 1\}, 1 \leq d \leq D$，），$D \geq 2$）定义

$$G_i(R) = G_{i_D}(\cdots(G_{i_d}(\cdots(G_{i_1}(R))\cdots))\cdots),$$

进一步，使 $\mathrm{M} = \prod_{d=1}^{D} \mathrm{M_d}$，对于 $R \in I_N, i \in \mathrm{M}$，函数 $f_R: \mathrm{M} \to I_N$ 定义如下。

$f_R(i) = G_i(R)$

用多维坐标法，对于任意的 $i \in \mathrm{M}$ 可以构筑

坐标的维数：D

各维的大小：$M_1,\ldots,M_D$

产生和管理 key 的总数：$M = M_1 \times \cdots \times M_D$ 的 key 管理系统 $R_i = f_R(i)$

并且，为了使讨论单纯化，本文使 $m = M_d$，而对于 G，只要 $m \geq 2$，就可以用提案法构

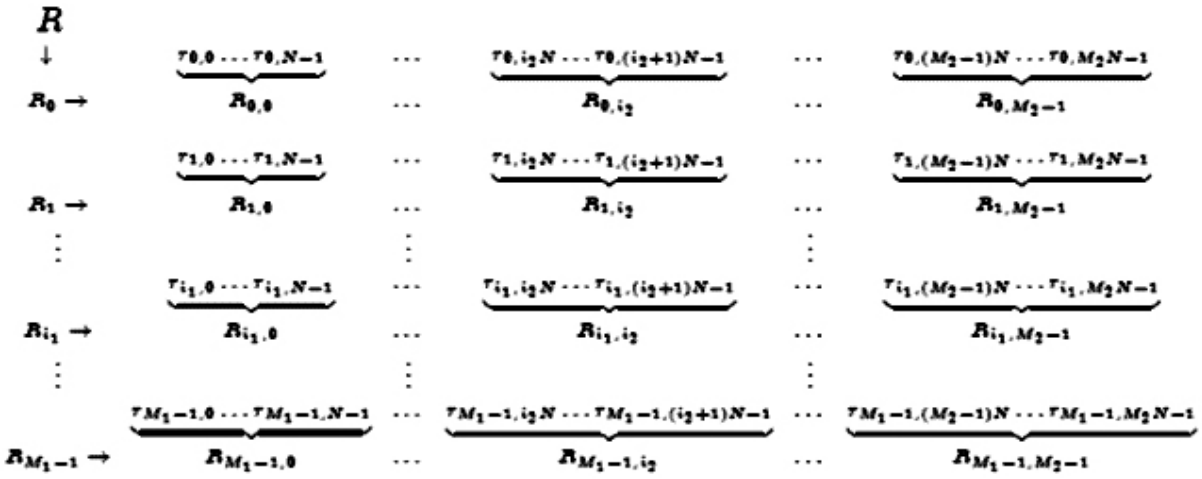

图 2. key 的二维配置

筑 key 管理系统。

图 2 显示 $D=2$ 时，利用提案法的 key，$R_i = f_R(i)$ 的产生例。此时，坐标 i 是以 2 个部分坐标对 (i_1,i_2) 来表现。

首先，由种子 R 产生的 M_1 个的种子 $R_0,\ldots,R_{M_1-1}$ 中选择 R_{i_1}。然后，由种子 R_{i_1} 产生的 M_2 个的 key, $R_{i_1,0},\ldots,R_{i_1,M_2-1}$ 中选择 R_{i_1,i_2} 输出。在这个例里，要注意的是维数 D 时种子要更新 D-1 次。

尚且，$m=1$时，从多维坐标法的构成，我们很容易知道无法用多维坐标法来构筑 key 的管理系统。

3.2 key 的产生时间

维数为 D 的多维坐标法，根据位置坐标 $i=(i_1,\ldots,i_D)$，由 N 比特的种子 R 来产生 key R_i。因此，伪随机数产生器 G 的产生 2 值系列的速度为 Sbps（比特 / 秒）时，任意的 N 比特的 R_i的产生所需的时间 T_{R_i} 由

$$T_{R_i} = (D+i_1+\cdots+i_D)\times N/S$$

来计算。在此，我们显示维数 D，总数$M = M_1\times\cdots\times M_D$ 的 N 比特的 key 的产生所需要的时间的最小值，最大值，平均值各为，$T^D_{\min}$ $T^D_{\max}$, T^D_{ave} 时的值。很明显，最小值是 $i_1,\ldots,i_D$ 都为最小值 $(0,\ldots,0)$，最大值是 $i_1,\ldots,i_D$ 都为最大值 $(M_1-1,\ldots,M_D-1)$ 时，平均值是由所有的 R_i 产生时间的和除以其总数 M 来计算。因此，各值如下。

$$T^D_{\min} = ND/S$$

$$T^D_{\max} = N(M_1+\cdots+M_D)/S$$

$$T^D_{ave} = \frac{N}{\prod_{j=1}^{D} M_j}\sum_{i_1=0}^{M_1-1}\cdots\sum_{i_D=0}^{M_D-1}(D+i_1+\cdots+i_D)/S$$
$$= N(D+M_1+\cdots+M_D)/(2S)$$

3.3 适当的维数 D

适当的维数 D 的值，可以由使 key 的产生所需要的时间的平均值 T^D_{ave} 最小化来求得。在这里，为了使最小化问题的解决更容易，使各维的大小 M_d都为一样的值。由此，可得到

$$T^D_{ave} = ND(1+M^{1/D})/(2S)$$

由于 N，S，M 为定数，T^D_{ave} 为最小值时，其导函数为 0，由此，可得到

$$1+M^{1/D}-M^{1/D}\ln M^{1/D}=0$$

因此可得到

$$3.59<M^{1/D}<3.60$$

由于各维的大小 $M_d = M^{1/D}$为整数，因此得来的 T^D_{ave} 不是最小值，但是，可以知道 $M^{1/D}=4$ 时是最接近最小的 T^D_{ave}的值。由此，多维坐标法的，管理的 key 的总数为 $M = M_1\times\cdots\times M_D, M_1=\cdots=M_D$ 时的贴切的维数 D，以速度优先时，由 $M^{1/D}=4$ 得到

$$D=\log_2 M^{1/2}$$

来计算。

3.4 多维坐标法的安全性

定理 1

对于在 3.1 定义的 $f_R(i)=G_i(R)$ ，使 $F_D=\{f_x\}_{x\in I_N}$ ，$F=\{F_D\}$，如果 G 是 CSB 产生器，F 合格于所有的多项式时间的统计学上的检验。

证明 .

省略（文献 3）。

定理 2

对于在 3.1 定义的 $f_R(i)=G_i(R)$ ，使 $F_D=\{f_x\}_{x\in I_N}$ ，$F=\{F_D\}$，设存在计算 $f_R(i)$ 的多项式程序，当 F 合格于所有的多项式时间的统计学上的检验时，并且，仅限于此时，F 是无法推测的。

证明 .

省略（文献 3）。

根据定理 1 和 2，多维坐标法满足所使用的伪随机数产生器 G 是 CSB 产生器，并具有一方向性，的条件，就是安全的。

3.5 效果

我们显示将 2 节介绍的整数 logistic map 随机数产生法作为 G（ N=128）时，用多维坐标法来产生 key 的所需时间。G 的伪随机数的产生速度为 749 Mbps 时，对维数 D=30，M_d=4，和 D=15，M_d=16 时的一个 key 的产生所需的时间的最小，最大，平均值如表 2。其中计算值为利用上述计算式计算的结果，实测值为用计算机实际计算的结果。

表 2 多维坐标法的 key 的产生时间 (单位 : μ sec)

	(D, M_d)=(30, 4)		(D, M_d)=(15, 16)	
	计算值	实测值	计算值	实测值
Min	5.13	5.89	2.56	2.98
Max	20.51	21.75	41.01	42.61
Ave	12.82		21.79	

Intel Pentium M processor 1.60 GHz

可见，当维数 D=30，Md=4 时我们可以以小于 22 μ sec 的时间，产生 2^{60} 个中的任意一个 key。

并且，我们可以通过管理多维坐标 $i_1,\ldots,i_D$ 来管理 key R_i。而 $i_1,\ldots,i_D$ 是无需保密的。需要严密管理（保密）的只有一个 R。亦就是，我们只要保障 R 的安全，就能保证 key R_i 的安全。$i_1,\ldots,i_D$ 则可以公开管理。

4 结言

我们将 2 篇论文 [1，3] 合并在一起作介绍的原因是由于，虽然整数 logistic map 法比其他产生法有着显著的优点，但是，仅仅靠整数 logistic map 法还无法实现流加密和一码一认证。也就是其优点无法体现在实际应用中。而对多维坐标法来说，利用整数 logistic map 法作为 G，并不是唯一选择，但显然是最佳选择。

整数 LMAP 的最主要的计算是乘法，而在计算机原理上属于较为复杂的乘法计算，在实际的计算机上，比如，32 比特的 CPU 上，32 比特的整数的乘法和加法的计算时间是一样的。由此，我们可以充分利用乘法器的计算能力对所需精度 N 进行适当的分割计算，来实现 N 比特的整数 LMAP 的计算。而即使是比较小的单片机，也都装备有乘法器，因此，我们可以在任何的计算机上实现相对高速的整数 LMAP 的计算，来产生安全的伪随机数。在专用芯片上，用小乘法器并列计算的方法，可以在时钟频率较低的条件下，也能高速产生随机数。提高了实现国产化的可能性。

多维坐标法利用整数 LMAP 法作为 G，使得其也能在任何的计算机上实现相对高速的 key 的产生，实现对 key 的管理，以实现一钥一密和一码一认。多维坐标法和整数 LMAP 法相辅相成，使各自的效能得到最大的实现。

因此，我们期待，整数 LMAP 法和多维坐标法可在安全领域，至今被认为困难的，对窃听有效的一次性码的认证系统，破解困

难的一次性密钥的加密系统，对违法复制的识别有效的将无法推测的识别码赋予每个产品及认证的系统等的构筑中得到广泛应用。

参考文献

[1] Jiguo DONG and Hiroyoshi MORITA, “Random Numbers Generation by Means of Integer Logistic Map and Mixing Operation”, IEICE vol.J94-A, no.12, pp.923-931, Dec. 2011(japanese)

[2] Jiguo DONG and Hiroyoshi MORITA, “Various Characters of Integer Logistic Map : Divergence, Convergence, and Periodicity”, IEICE vol.J96-A, no.2, pp.90-99, Feb. 2013. (japanese)

[3] Jiguo DONG and Hiroyoshi MORITA, “A Multi-Dimensional Index Method for Key Generation and Key Management”, IEICE vol. J98-D, no.1, pp.206-213, Jan. 2015. (japanese)