Europäisches Patentamt
European Patent Office
Office européen des brevets

URKUNDE

Es wird hiermit bescheinigt, dass für die in der Patentschrift beschriebene Erfindung ein europäisches Patent für die in der Patentschrift bezeichneten Vertragsstaaten erteilt worden ist.

CERTIFICATE

It is hereby certified that a European patent has been granted in respect of the invention described in the patent specification for the Contracting States designated in the specification.

CERTIFICAT

Il est certifié qu'un brevet européen a été délivré pour l'invention décrite dans le fascicule de brevet, pour les Etats contractants désignés dans le fascicule de brevet.

Europäisches Patent Nr. / European patent No. / Brevet européen nº

2309381

Patentinhaber / Proprietor of the patent / Titulaire du brevet

Dong, Jiguo
401, 2-4-5 Misakicho
Chiyoda
Tokyo 101-0061/JP

München, den / Munich, / Fait à Munich, le 04.11.15

Benoît Battistelli
Präsident des Europäischen Patentamts
President of the European Patent Office
Président de l'Office européen des brevets

EPA/EPO/OEB 2031 07.10

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 309 381 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
04.11.2015 Bulletin 2015/45

(21) Application number: **09802795.6**

(22) Date of filing: **09.06.2009**

(51) Int Cl.:
G06F 7/58 *(2006.01)* ***G09C 1/00*** *(2006.01)*
H04L 9/26 *(2006.01)*

(86) International application number:
PCT/JP2009/060815

(87) International publication number:
WO 2010/013550 (04.02.2010 Gazette 2010/05)

(54) **RANDOM NUMBER GENERATION AND MANAGEMENT METHOD, AND DEVICE**

VERFAHREN UND VORRICHTUNG ZUR ZUFALLSZAHLENERZEUGUNG UND -VERWALTUNG

PROCÉDÉ ET DISPOSITIF DE GÉNÉRATION ET DE GESTION DE NOMBRE ALÉATOIRE

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK TR

(30) Priority: **28.07.2008 JP 2008213305**

(43) Date of publication of application:
13.04.2011 Bulletin 2011/15

(73) Proprietor: **Dong, Jiguo**
Tokyo 101-0061 (JP)

(72) Inventor: **Dong, Jiguo**
Tokyo 101-0061 (JP)

(74) Representative: **Hellmich, Wolfgang**
European Patent and Trademark Attorney
Lortzingstrasse 9 / 2. Stock
81241 München (DE)

(56) References cited:
JP-A- 9 244 876 **JP-A- 2006 338 045**
JP-A- 2008 070 727

- **R Ursulean: "Reconsidering the Generalized Logistic Map as a Pseudo Random Bit Generator", T 191 AUKSTUJU DAZNIU TECHNOLOGIJA, 1 January 2004 (2004-01-01), XP055127555, Retrieved from the Internet: URL: http://www.ee.ktu.lt/journal/2004/7/Ursulean.pdf [retrieved on 2014-07-08]**
- **D. SORNETTE ET AL: "Chaos, pseudo-random number generators and the random walk problem", JOURNAL DE PHYSIQUE, vol. 45, no. 12, 1 January 1984 (1984-01-01), pages 1843-1857, XP055127564, ISSN: 0302-0738, DOI: 10.1051/jphys:0198400450120184300**

EP 2 309 381 B1

Description

BACKGROUND OF THE INVENTION

1. Field of Invention

[0001] The present invention relates to a random number generation and management method and device.

2. Description of the Related Art

[0002] There is not only a situation of a large number of random numbers generated and used in a short time during computer simulation, but also a situation of random numbers with finite length (such as 128 bits) generated and used in identification (ID) and password (PW). Even ID and PW require stronger security management, in fact, there is not yet a device that achieves effective management. There are often reports related to lost memory media storing user's ID and other information.

[0003] The management for the random numbers array of ID and PW is difficult, and currently the information is only stored in the storage of memory media. User's ID and other information stored in the memory media being lost under the state without encryption shows the difficulty of the management.

[0004] Pseudo-random numbers can be regenerated. When the same random number is needed during computer simulation, it can be generated if the initial value generating the random numbers is kept. That is, the management for the initial value generating the random numbers is regarded as the management for the very long random number series.

[0005] On the premise of a sufficiently long periodic binary series ($r_0 r_1 \ldots r_t \ldots$) (discrete-time t=0, 1, 2, ...), we consider the following ID and PW generating and managing system. ID and PW system is generated and managed through an initial value by secret management and t by public management. When needed, the initial value and t are input in the system to generate the corresponding ID or PW. But, it is clear that it takes a long time to generate random number series (ID, PW) corresponding to large t, and the system will lose its practical function, such as $t=2^{64}$.

[0006] On the other hand, because the value series generated by the chaos function has special properties, such as nonlinearity, initial value sensitivity, calculation unidirectionality, etc, it is expected to be used for random numbers generation. In which, some are used for generating random numbers through logistic map (Equation 1) (hereinafter referred to as LMAP), and for inspection to the generated series, (non-patent literatures 1-3).

Equation 1

$$x_{t+1} = 4x_t(1-x_t)$$
$$(0 < x_t < 1, \quad t = 0,1,2,\ldots)$$

Non-patent literature 1

[0007] Ulam, S.M. and Von Neumann, J., "On Combination of Stocastic Determistic Processes", Bull. AMS., Vol. 53, p.1120 (1947)

[0008] In this literature, Equation 1 is proposed to be used for generating random numbers.

Non-patent literature 2

[0009] Tohru KOHDA and Eiji OGATA, "Bernoulli trials and Chaotic Trajectories in the Logistic map", IEICE A (Japanese), Vol. J68-A No.2 pp. 146-152 (1985)

[0010] In this literature, the binary series generated through Equation 1 and the threshold value defined as 0.5 is complete random number series.

Non-patent literature 3

[0011] K. Shono, "Chaos engineering", Springer-Verlag Tokyo, Tokyo, 2002.

[0012] In this literature, Equation 1 applies to an effective method for high-speed random numbers generated, in which fixed decimal point calculation is proposed to implement hardware-based.

[0013] The calculation of chaos function LMAP has unidirectionality, that is, starting from a certain initial value x_0,x_t (t=1,2,...) can be calculated individually, but x_0 cannot be calculated from the calculated x_t. This is because if the inverse

function of the second order function LMAP, $x_t = (1 \pm\sqrt{(1-x_{t+1})})/2$, is used for reverse calculation for x_t, one of the two possible states must be chosen (the choice between + or - symbol).

[0014] In the application, the binary series generated through the LMAP is the binary series generated when the threshold value is 0.5 to x_t, (if $x_t \geqq 0.5$, the output is 1, and if $x_t < 0.5$, the output is 0). Moreover, the inventor of the application confirms that the LMAP has the following features.

[0015] When the calculation accuracy is N bits, from t=0 to t=N-1, if there are continuously generated N bits binary series (the threshold value is 0.5), the x_0 generating the binary series can be calculated through the inverse function of the LMAP. The + or - symbol is chosen on the basis of the binary value corresponding to the same t (1: +, 0: -). This is because the LMAP is calculated according to the divergent results of the lyapunov exponent, and the inverse function of the LMAP is calculated according to the convergent results of the lyapunov exponent.

[0016] It is clear that the binary series generated by the LMAP with such features is improper to be used for the random numbers of information security. For the binary series generated by a very long LMAP, if the foremost bits longer than the calculation accuracy is known, through calculating the inverse function of the LMAP on the binary bits, x_0 is obtained, and through calculating the LMAP on the x_0, all bits of the very long binary series can be calculated. But, the upper degree bits (generated earlier) cannot be calculated through the lower degree bits (generated later) of the binary series generated through the LMAP.

[0017] If the calculation accuracy is N bits, the number of continuous 0's generated during the calculation of the LMAP is less than N/2. The feature limits the possibility of the combination number for the binary series generated by the LMAP, but simultaneously ensures that when the N bits are extracted from the binary series to make a new initial value, the new initial value does not occur the value called a black hole, such as 0, 0.25, 0.5, 0.75, etc.

[0018] When the LMAP is calculated in limited calculation accuracy, the initial value sensitivity can be observed in the following forms. Through two initial values, in which except for the lowest bit, the other bits in the two initial values are same, we can ensure that the internal states x_t are in completely different tracks after N times calculation (there is not any relationship between the two internal states), for example, the results of the initial value 0.0...01 and 0.0...010 after 128 times calculation in 128 bits calculation accuracy are respectively 0.0100...0100 and 0.1100...0110 (only upper degree 4bits and lower degree 4 bits are shown). There are two meanings.

[0019] Firstly, because of the calculation in limited calculation accuracy, the lower degree bits are cast out, thus there is no relationship between the state after N times calculation and the initial state, that is, the state after N times calculation, not been calculated, cannot be estimated.

[0020] Secondly, even if there are two initial values, in which except for the lowest bit, the other bits in the two initial values are same, there is no correlation between the binary series individually generated by the two initial values, and the content of one series cannot be estimated through the other series.

[0021] The features of the binary series generated through the above LMAP have important implications in the following method for generating multi-dimensional random numbers.

Non-patent literature 4

[0022] R. Ursulean, "Reconsidering the Generalized Logistic Map as a Pseudo Random Bit Generator", http://www.ee.ktu.lt/journal/2004/7/Ursulean.pdf.

[0023] This document discloses an Equation 1 which is transformed into Equation 4. In responding to a different β, they are examined by the binary series raised from Equation 4 (through Equation 2 or Equation 3) in Ursulean.

Equation 1 of Ursulean: $$x_{n+1} = r x_n (1 - x_n)$$

Equation 2 of Ursulean: $$b_n = \begin{cases} 0 & x_n \leq \bar{x} \\ 1 & x_n > \bar{x} \end{cases}$$

where $\bar{x}$ denotes the mean value and b_n is the bit generated by the n-th iteration of the map.

Equation 3 of Ursulean: $$b_n = \begin{cases} 0 & x_n \leq med \\ 1 & x_n > med \end{cases}$$

where *med* denotes the median of the values generated by the discrete chaotic map.

[0024] The value of x_n is between 0 and 1. That is when β =1, $med = \frac{0+1}{2} = 0.5$.

Equation 4 of Ursulean: $x_{n+1} = (\beta+1)\left(1+\frac{1}{\beta}\right)^{\beta} x_n(1-x_n)$ →Normalization

SUMMARY OF THE INVENTION

[0025] The object of the present invention is to provide a method and device for generating and managing random numbers, which is easily realized in general-purpose computer and special hardware, to generate and manage shorter random number series, such as ID and PW.

[0026] The method for generating and managing random number of the present invention (Claim 1) is for a given N, where N is an integer and N≧2, bits binary series R and K dimensions multi-dimensional coordinate information i (i1, ..., ik, ..., iK), where ik are integer and ik≧0, k:1, 2, ..., K, in which R is used for the decimal portion of initial value x_0 for nonlinear function, $X_{t+1} = 4X_t(1 - X_x)$ (herein after referred to as LMAP, $0 < x_t < 1$), i is transformed into coordinates of each dimension, i1, ..., ik, ..., iK, and x_0 and i1, ..., ik, ..., iK are stored in a register; the method with the features including:

1) for the initial value x_0 and ik stored in the register, a chaos computing unit repeatedly implements the calculation of the LMAP in N bits calculation accuracy through fixed decimal point calculation, to generate N bits binary series B_{ik}, and the bits of B_{ik} are individually constituted by $b_{k,0}, b_{k,1}$,... and $b_{k,N-1}$, wherein $b_{k,0} = [2 \times x_{N\times ik}]$, $b_{k,1} = [2 \times x_{N\times ik+1}]$, ... and $b_{k,N-1} = [2 \times x_{N\times ik+N-1}]$, [] represents a calculation of removing the decimal portion;

2) B_{ik} is then used for the decimal portion of initial value x_0 through the chaos computing unit repeatedly implementing the calculation of the LMAP to generate N bits binary series $R_{i1,...,ik}$, where $R_{i1,...,ik}$ is R_{i1} when k=1, and $R_{i1,...,ik}$ is $R_{i1,...ik}$ when k=K, and the bits of $R_{i1,...,ik}$ are individually constituted by $r_{k,0}, r_{k,1}$,... and $r_{k,N-1}$, wherein $r_{k,0} = [2 \times x_N]$, $r_{k,1} = [2 \times x_{N+1}]$, .., $r_{k,N-1} = [2 \times x_{2N-1}]$; and store the binary series $R_{i1,...,ik}$ in a random number register.

3) the $R_{i1,...,ik}$ stored in the above random number register is used for the decimal portion of initial value x_0 of the LMAP and stored in the above register. But $R_{i1,...,ik}$ do not perform the transformation of the initial value x_0 of the LMAP when k=K.

[0027] The above operations 1), 2) and 3) are implemented for each dimension k=1,2, ..., K, to implement the method for generating and managing random numbers with the feature of multi-dimensional random number R_i.

[0028] The device for generating and managing random numbers of the present invention are with the features of the following parts:

an input unit of generating and managing information for random number, which receives N , where N is an integer and N ≧ 2, bits binary series R as the initial value information to generate random numbers, and K dimensions multi-dimensional coordinate i (i1, ..., ik, ..., iK) where ik is an integer and ik ≧ 0, and K is an integer and K≧ 1 as the multi-dimensional coordinate information;

an initial value/multi-dimensional coordinates transformation unit, which transforms R, the decimal portion of initial value x_0 for nonlinear function, $x_{t+1} = 4x_t(1-x_t)$ (herein after referred to as LMAP, $0 < x_t < 1$), into x_0, and transforms I into multi-dimensional coordinates i1, ..., ik, ..., iK to prepare for the calculation of the LMAP to generate random numbers;

a register, which stores the initial value x_0 and multi-dimensional coordinates i1, ..., ik, ..., iK transformed by the initial value/multi-dimensional coordinates transformation unit;

a chaos computing unit generating chaos binary series, which repeatedly implements the calculation of the LMAP in N bits calculation accuracy through fixed decimal point calculation, on the basis of the initial value x_0 and multi-dimensional coordinates stored in the register, to generate N bits binary series B_{ik}, the bits of B_{ik} are individually constituted by $b_{k,0}$, $b_{k,1}$,... and $b_{k,N-1}$, wherein $b_{k,0} = [2 \times x_{N\times ik}]$, $b_{k,1} = [2 \times x_{N\times ik+1}]$, ...and $b_{k,N-1} = [2 \times x_{N\times ik+N-1}]$, and [] represents an operator used to remove decimal portion; and B_{ik} is used for the decimal portion of initial value

x_0 and transformed into x_0 through the chaos computing unit repeatedly implementing the calculation of the LMAP to generate N bits binary series $R_{i1,...,ik}$, where $R_{i1,...,ik}$ is R_{i1} when k=1, and $R_{i1,...,ik}$ is $R_{i1,...ik}$ when k=K, and the bits of $R_{i1,...,ik}$ are individually constituted by $r_{k,0}$,$r_{k,i}$, ... and $r_{k,N-1}$, wherein $r_{k,0} = [2 \times x_N]$, $r_{k,1} = [2 \times x_{N+1}]$, ... , $r_{k,N-1} = [2 \times x_{2N-1}]$;

a random number register, which stores N bits binary series $R_{i1, ...,ik}$ output by the chaos function computing unit; and

a random number generation control unit, which implements the following operations, including an operation of the input unit of generating and managing information for random numbers, an operation of the initial value/multi-dimensional coordinates transformation unit, a calculation of the chaos function computing unit, and a calculation of x_0 transformed from $R_{i1,...ik}$, as the decimal portion of the initial value x_0 of the LMAP, for each dimension k=1,2, ..., K, but $R_{i1,...,ik}$ do not perform the transformation of the initial value x_0 of the LMAP when k=K, to generate multi-dimensional random number R_i.

[0029] Random number series in the present invention can be quickly generated (regenerated) to be placed at any coordinate position among plenty of random number series in multi-dimensional coordinate space. Therefore, as shown in FIG. 1, we can construct a system for generating and managing random numbers, to generate and manage binary series R_i (ID, PW) through managed binary series R and regulated multi-dimensional coordinate information i (i1, ···, ik, ···, iK). Because of the chaos function (LMAP) used in the present invention, there is no linear relationship between the R_i generated by regulated multi-dimensional coordinate information i (i1, ···, ik, ···, iK). Thus, R cannot be estimated through certain random number series R_i and multi-dimensional coordinate information i (i1, ···, ik, ·· ·, iK) thereof. Also, the random number series generated by other multi-dimensional coordinate information cannot be estimated.

[0030] Still because of the LMAP calculated through fixed decimal point calculation in the present invention, the system for generating and managing random numbers proposed by the present invention is easily constructed by a system implementing integer arithmetic, (such as multiplication, addition, subtraction, and logical operations) whether it is a software or hardware system. Furthermore, the present invention makes the low price of the system for generating and managing random numbers possible to be an expected form in industrial technology.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031]

FIG. 1 is a schematic view of a system for generating and managing random numbers according to the present invention;

FIG. 2 is a composition diagram showing the implementation form of a device for generating random numbers according to the present invention;

FIG. 3 shows the method for generating one- dimensional random numbers;

FIG. 4 shows the method for generating two-dimensional random numbers;

FIG. 5 is a flowchart showing the embodiment of the action of the device for generating and managing a random number 100;

FIG. 6 is a table showing the product numbers of industrial products transformed into multi-dimensional coordinates; and

FIG. 7 is a table showing the time needed for the random numbers generated by the method for generating multi-dimensional random numbers.

DETAILED DESCRIPTION OF THE INVENTION

[0032] Hereinafter, the embodiments and the effects of the present invention will be described with reference to the drawings.

[0033] FIG. 2 shows an embodiment of the present invention, which is the device for generating and managing a random number 100. The device for generating and managing the random number 100 in the embodiment generates N bits binary series $R_{i1,...,ik}$ by way of chaos computing unit, through transforming N bits binary series R into initial value

x_0 of the LMAP, and on the basis of x_0 and multi-dimensional coordinate information i, and then the calculation of transforming $R_{i1,...,ik}$ into x_0 is repeatedly implemented for K times, to generate K-dimensional random number R_i. The device for generating and managing the random number 100 is instituted by an input unit of generating and managing information for random number 102, an initial value/multi-dimensional coordinates transformation unit 104, a register 106, a chaos function computing unit 108, a random number register 110, and a random number generation control unit 112.

[0034] The input unit of generating and managing information for random numbers 102 receives N bits binary series R and multi-dimensional coordinate information I (i1, ..., iK).

[0035] The initial value/multi-dimensional coordinates transformation unit 104 transforms the input binary series R or the binary series $R_{i1,...,ik}$ generated during the generation of the multi-dimensional random numbers into x_0, and transforms the input multi-dimensional coordinate information i into each dimension coordinates i1, ..., iK for fixed decimal point calculation to generate multi-dimensional random numbers.

[0036] The register 106 stores x_0 and i1, ..., iK transformed in the initial value/multi-dimensional coordinates transformation unit 104.

[0037] The initial value stored in the register 106 is used by the chaos function computing unit 108 for each dimension of multi-dimensional coordinate ik (k=1, 2, ..., K) to repeatedly calculate the LMAP, to generate N bits binary series $R_{i1,...,ik}$.

[0038] The detailed operations of the chaos function computing unit are described as following.

[0039] Firstly, the initial value x_0 and multi-dimensional coordinate ik stored in the register 106 are used by the chaos function computing unit 108 to calculate the LMAP for N×ik times.

[0040] Next, while in the calculation of the LMAP, $b_{k,0}=[2 \times x_{N\times ik}]$, $b_{k,1} = [2 \times x_{N\times ik+1}]$, ...and $b_{k,N-1} = [2 \times x_{N\times ik+N-1}]$ are calculated, in which [] represents a calculation of removing decimal portion, to generate N bits binary series $B_{ik}(b_{k,0}, b_{k,1}, ..., b_{k,N-1})$.

[0041] Then, B_{ik} is used for the decimal portion of initial value x_0 of the LMAP, to calculate the LMAP for N times.

[0042] And then, while in the calculation of the LMAP, $r_{k,0} = [2 \times x_N]$, $r_{k,1} = [2 \times x_{N+1}]$, ... and $r_{k,N-1} = [2 \times x_{2N-1}]$ are calculated, to generate N bits binary series $R_{i1,...,ik}(r_{k,0}, r_{k,1}, ..., r_{k,N-1})$.

[0043] Because of the above additional calculation, we can cut off the above-mentioned possibility of speculating the lower degree bits from the upper degree bits in the binary series generated through the LMAP.

[0044] The random number register 110 is used for storing the binary series output by the chaos function computing unit 108.

[0045] The random number generation control unit 112 controls the operations of every unit to generate multi-dimensional random number series.

[0046] The random number generation control unit 112 repeatedly calculates on the basis of K dimensions multi-dimensional information for each dimension of k=1 to k=K, through the chaos function computing unit 108, and the outputs are stored in the random number register 110; if k<K, the random number series stored in the random number register 110 is transformed into x_0 through the initial value/multi-dimensional coordinates transformation unit 104, and the x_0 is stored in the register 106, and the calculation is gone on; if k=K, the N bits binary series stored in the random number register 110 is used for the random number R_i, and the calculation is terminated.

[0047] The meaning and effect of the multi-dimensional random number are described as following.

[0048] We consider the fixed decimal point calculation to the LMAP with N bits calculation accuracy, to generate N bits binary series.

[0049] N bits initial value x_0 is input to generate the required number of binary series $r(r_0, r_1, ...)$. We assume that the generation speed of the random number generator is S bps (bits/sec), and the operational time, except for generating binary series, is too short, so not be counted. Here we take a look at that, if the random number generator generates MxN, where M is an integer and M≧1, bits random numbers, the required time for generating any one of R_i through initial value x_0, when the random number is divided into M series and each one is N bits binary series.

[0050] The series R_i generated through this way can be imagined to be put on a straight line (as shown in FIG. 3), thus, the generation method is referred to a method for generating one dimension random number series.

[0051] It is clear that, because of N bits series, the required time for generating the first series Ro through x_0 is N/S. The required time for generating the second series R_1 through x_0 is 2N/S, because it includes the time for generating the first random number series. Thus, the required time for generating the ith series R_{i-1}, through x_0 is ixN/S.

[0052] Because the more generated random number series, the better, a greater M is expected. But, as for such method for continuously generating random number series through one initial value, M cannot be very large, such as $M=2^{64}$, even if the generation speed is 2^{40} bps, the average generating time for N bits series is $(1+2^{64}) \times N/2^{40} > 2^{24}$ seconds.

[0053] We will consider the following method for generating random numbers. Here, we assume the M is $M_1 \times M_2$ where M_1, M_2 are integers and M_1, $M_2 \geqq 1$, M_1 series N bits random number series are generated through N bits initial value x_0, the generated M_1 series N bits random number series are used for new initial values, to individually generating M_2 series N bits random number series. The generated N bits random number series are represented as $R_{i1,i2}$ (i1 =

0, ...,M_i-1, $i2$ = 0, ..., M_2 -1).

[0054] The series $R_{i1\cdot i2}$ generated through the above way can be imagined to be put on a two-dimensional plane (coordinates), thus, the generation method is referred to a method for generating two-dimensional random number series.

[0055] It is clear that the total number of $R_{i1\cdot i2}$ is $M_1 \times M_2$=M, the required time T for generating any one $R_{i1\cdot i2}$ through initial value x_0 is as following.

$$T_{R_{0,0}} = (1 \times N + 1 \times N)/S = 2N/S$$

$$T_{R_{0,1}} = (1 \times N + 2 \times N)/S = 3N/S$$

$$T_{R_{0,i2}} = (1 \times N + (i2+1) \times N)/S = (i2+2)N/S$$

$$T_{R_{i1,i2}} = ((1+i1) \times N + (i2+1) \times N)/S = (i1+i2+2)N/S$$

[0056] That is, as for the required time for generating one random number series, the minimum is 2N/S, and the maximum is (M_1+M_2) N/S, compared with the longest time, M×N/S= ($M_1 \times M_2$) N/S, for the method for generating one dimension random number series, the difference is self-evident.

[0057] We assume that K dimensions and M=$M_1 \times ... \times M_K$, M_1 series N bits random number series are generated through N bits initial value x_0. Then, the generated M_1 series N bits series are used for new initial values, to individually generate M_2 series N bits random number series. Till M_K, the transformations for new initial values are performed for K-1 times to generate series R_i. The series R_i can be imagined to be coordinates (i: i1, i2, ...,iK) put in K dimensions space, thus, the generation method is referred to a method for generating K dimensions random number series. We can understand that, the required time for any one series in K dimensions coordinate space is (K+i1+i2+...+ iK) N/S, the minimum is KN/S, the maximum is (M_1+M_2+...+M_K) N/S, and the average time is (K+M_1+M_2+...+M_K) N/ (2S).

[0058] The meanings of every parameter N, K (k), M (Mk) in the method for generating multi-dimensional random numbers are described as following.

[0059] N is calculation accuracy. Therefore, the variety of initial value (initial value space) available is 2^N-4, except for 0....0, 010...0, 10...0, 110...0 (0, 0.25, 0.5, 0.75). But, because the length of chaos state generated through certain initial value x_0 is similar to $2^{N/2}$, we can estimate the value of non-periodic M is $2^{N/2}$/N. Actually, when N is assumed to be a value between 32 and 64, the search result through numerical calculation to the length of non-periodic M is similar to $2^{N/2}$. This is because the generation of the initial values through repeatedly calculation makes the periodicity of multi-dimensional random number generator to be longer than that of the original LMAP.

[0060] The coordinate volume M is the total of the series generated through initial value x_0. The value is generally restricted by the periodicity of random number generator. However, if same random number series is allowed being placed in multi-dimensional space (an unspecific location), there is no such restriction.

[0061] The space of every dimension coordinate depends on M_k (k=1, 2, ..., K). Also, the generation speed of R_i depends on M_k. When M_k is assumed to be larger, the average generation speed of R_i would be much slower.

[0062] M_k and k together have the classification function to R_i. Namely, the information with specific meanings (such as time, name, organization (department), etc.) can be possessed. That is, when the decision for the appropriate value of M_k is made, not only generation speed of random numbers, but also classification ease, must be thought over. The more effective representing way for coordinate space is assumed $M=2^m$, $m=m_1+...+m_K$ where $m_1,...,m_K$ are integers and $m_1,...,m_K \geqq 0$.

[0063] K is used for dimension number, possessing classification function. When M is constant, if K is larger, the average value of M_k will be smaller, and the average generation speed of R_i will be faster. However, if the individual M_k is extremely larger, generation speed of R_i may be extremely slower. This is because, like the method for generating one dimension random number series, if the coordinate number in certain dimension is assumed too large, the average generation time in this dimension will be very long. Therefore, the setting of K is not only following the classification function, but also making individual M_k not too large. That is, even of the same classification, there is no need to represent as the same dimension coordinate. In the classification, when the individual M_k is too large, it can be represented as plural dimensions.

K (k) can be divided.

[0064] We will think about a multi-dimensional random number generation system, including N bits calculation accuracy, initial value x_0, and K dimensions, and the magnitude of every dimension is M_1, M_2, ..., M_K. We inspect the state of k dimension in the intermediate process.

[0065] When k=1, M_1 N bits binary series R_{i1} are generated through initial value x_0. When k=2, M_1 N bits binary series R_{i1} (initial value in the intermediate process) are generated through initial value x_0, then, M_2 N bits binary series $R_{i1, i2}$ are individually generated through initial value in the intermediate process. Namely, total $M_1 \times M_2$ N bits binary series can be generated through initial value x_0. Also, the k dimension in intermediate process can generate $M_1 \times M_2 \times ... \times M_k$ initial values in intermediate process.

[0066] Here, we assume k=Ku and K=Ku+Kd, then Kd=K-Ku. That is, which is the multi-dimensional random number generation system constituted by the upper degree dimension of initial value x_0 with Ku dimensions and the values of every dimension are M_1, M_2, ..., M_{ku}. For example, $M_1 \times M_2 \times ... \times M_{ku}$ N bits binary series (the initial value in intermediate process) generated through the system constituted by the upper degree dimension of Ku dimensions are used for the multi-dimensional random number generation system (Kd dimensions and the values of every dimension are M_{ku+1}, M_{ku+1}, ..., M_K) of every initial value, which is constituted by the lower degree dimension.

[0067] As mentioned above, two K's (Ku, Kd) correspond to a multi-dimensional random number generation system, through dividing K, the multi-dimensional random number occurs system-division, and when the multi-dimensional random number generation system is divided, through Ku dimension as the margin, not all lower degree dimensions should be divided from the multi-dimensional random number generation system. Only the random number $R_{i1,i2,...,iku}$ at the arbitrary coordinates i1, i2, ..., i (ku) in intermediate process are output to be the initial value of the new lower degree multi-dimensional random number generation system, arbitrary lower degree multi-dimensional random number generation system can be divided from the original multi-dimensional random number generation system.

[0068] Furthermore, it is clear that the inverse operation of the system-division should connect (integrate) with the divided system. However, the individual coordinate information in intermediate process dimension during division should be kept. In other words, the intermediate dimensional coordinate information of the divided system with the need of re-connection should be kept. The intermediate process coordinate of completely independent divided system without the need of re-connection may be removed.

[0069] The above describes the division for K into Ku and kd, and, of course, more division can be performed as needed through same method.

[0070] The value of R_i generated from multi-dimensional coordinates i (i1, i2, ..., iK) through initial value x_0 and that of R_i generated from multi-dimensional coordinates i (i (ku+1), i (ku+2), ..., iK) through initial value $R_{i1,i2,...,iku}$ are same. That is, initial value $R_{i1,i2,...,iku}$ can be obtained from initial value x_0, but initial value x_0 cannot be obtained from $R_{i1,i2,...,iku}$. As for the above initial value with such relationship, initial value x_0 is referred to upper degree initial value, and $R_{i1,i2,...,iku}$ is referred to lower degree initial value.

[0071] By way of the multi-dimensional random number generation method with the above features, we can construct random number generation system with grade structure. That is, the competence of the upper degree initial value x_0 and that of the lower degree initial value $R_{i1,i2,...,iku}$ are different, the keeper of initial value x_0 has greater competence than that of $R_{i1,i2,...,iku}$.

[0072] FIG. 5 is a flowchart showing an embodiment of the action of the device for generating and managing the random number 100. The device for generating and managing the random number 100 shown in the flowchart starts action according to the order generated by the random number.

[0073] The input unit of generating and managing information for random numbers 102 receives the input N bits binary series R and multi-dimensional coordinate information i, and variable k is assumed as k=0. Then, the initial value multi-dimensional coordinates transformation unit 104 makes k=k+1, and transforms the received N bits binary series R into x_0 (S104) and stores x_0 and multi-dimensional coordinate information (i1, i2, ···, iK) into the register 106 (S106). According to x_0 and multi-dimensional coordinate information ik stored in the register 106, the chaos function computing unit 108 further generates N bits binary series $R_{i1,i2,...,ik}$, and stores them into the random number register 110 (S108); when k < K (dimension number), N bits binary series $R_{i1,i2,...,ik}$ are input the initial value/multi-dimensional coordinates transformation unit 104 (S110), and re-calculation is performed from (S104). When k=K, N bits binary series $R_{i1,i2,...,ik}$ are output as random numbers (S112), and the operation of the device for generating and managing random number 100 in the flowchart terminates.

[0074] Here, by way of the embodiment of the system for generating and managing random numbers, the effect of the present invention is confirmed through initial value sensitivity, irrelevancy of adjacent coordinates, generation speed, and system-division.

[0075] Industrial products include the identification of the product mark (number). For example, model number, manufacture number, batch number, manufacture date, etc, these numbers are necessary for managing related information of the product and after-sales service. Of course, these numbers must be managed. Most of the numbers of products

are regular consecutive numbers. Because of regularity, the message content in these numbers is less and easy to be managed. However, on the other hand, it is easy to infer other numbers from a number and to be illegally used, such counterfeit attached with authentic number to pose as genuine article. If the regular number is integrated with irregular identification ID number (random number), the product number cannot be inferred. However, using the irregular random number will make it difficult to manage near-infinite numbers of industrial products. The multi-dimensional random number generation method proposed in the present invention is effective to manage the irregular identification ID (random number series).

[0076] We assume three elements (R , i , R_i) corresponding to multi-dimensional random number generation method, which transform (unique) product number identifying product into multi-dimensional coordinate, and integrate the generated product identification ID (random number R_i) with the product number to make it non-inferable. At this point, initial value x_0 (R) becomes the management key for managing product identification ID. The correct initial value x_0 and product number can generate (regenerate) corresponded product identification ID. The regular product number can be inferred, but the attached product identification ID cannot be. That is, without initial value x_0, it is impossible to obtain correct combination of product number and product identification ID.

[0077] The following is an example of simplified generation for identification ID of the product with model number MC780 and manufacture number C042875.

[0078] The generation for identification ID needs to transform the product number into multi-dimensional coordinate. As shown in FIG. 6, the transformation for model number and manufacture number is performed. (The English letters are represented as hexadecimal in ASCII code, and decimal digits are represented as hexadecimal.)

[0079] We assume dimension number K=14, the values of every dimension are $M_1=M_2=...=M_{14}=16$, and management key is 128 bits initial value x_0 =0...0 1 (hexadecimal), thus, R_i calculated through multi-dimensional random number generation method is represented in hexadecimal as B44768B06B7A25D464F3523552BD0DFB. The corresponded multi-dimensional coordinate is represented in hexadecimal as (i=4,D,4,3,2,C,4,4,3,0,A,7,7,B).

[0080] Firstly, multi-dimensional coordinate i is fixed, and x_0 is given minor changes, to identify the generated changes of R_i. The following shows the embodiment when multi-dimensional coordinate i is (4,D,4,3,2,C,4,4,3,0,A,7,7,B). When x_0 is 0...01 and 0...02 (32 digits in hexadecimal), R_i is (B44768B06B7A25D464F3523552BD0DFB) and (8849A5994E6b861 F6298CFB4C7C71 E9F), respectively. We can confirm the initial value sensitivity.

[0081] Then, initial value is fixed, and multi-dimensional coordinate i is given minor changes, to identify the generated changes of R_i. The following shows the embodiment when initial value x_0 is 0...01. When multi-dimensional coordinate i is (4,D,4,3,2,C,4,4,3,0,A,7,7,B) and (4,D,4,3,2,C,4,4,3,0,A,7,7,C), R_i is (B44768B06B7A25D464F35235 52BD0DFB) and (BA1359675D951215D6F14411426D1E13), respectively. We can confirm the irrelevancy of adjacent coordinates.

[0082] From the above results, we can confirm the following effects; as for the random number generator constructed through multi-dimensional random number generation method, even if the system configuration is same, there is no relationship between the random numbers generated through different initial values and same multi-dimensional coordinate. That is, even the system are constructed the same, it is impossible to infer the initial value from the random number of the same multi-dimensional coordinate. Also, from the random number series at certain multi-dimensional coordinate, it is impossible to infer the random number series at the adjacent multi-dimensional coordinate.

[0083] As shown in the above, when the generation speed of binary series is given, the required time for generating the random number series at certain multi-dimensional coordinate is obtained. At this point, we assume that the operation time, except for generating binary series, is too short, so not be counted. Here, the generation time calculated through approximation is inferred time t, the actual required time for generation is example time t', and when the generation speed of binary series is 4.9Mbps, the results of the both are shown as in FIG. 7. We can confirm which are close and the generation speed can be practically applied.

[0084] In the system for generating product identification ID, the product number is instituted by model number and manufacture number, that is, the product number is divided into a portion for managing model (model number) and a portion for managing the product of the model (manufacture number). Here, the product number is divided into model number and manufacture number, to confirm the random numbers respectively generated in divided and undivided situations.

[0085] When initial value x_0 is 0...01 and multi-dimensional coordinate i is (4,D,4,3,2,C,4,4,3,0,A,7,7,B), the random number series R_i is generated as shown in the above. Firstly, the multi-dimensional coordinate is divided into (4,D,4,3,2,C,4) (model management) and (4,3,0,A,7,7,B) (the product of the model). Then, the random number series R_i', is generated through initial value x_0 (0...01) and multi-dimensional coordinate i' (4,D,4,3,2,C,4). And then, the random number series R_i'' is generated through the initial value transformed from $R_{i'}$ and multi-dimensional coordinate i'' (4,3,0,A,7,7,B), and compared with the random number series R_i when undivided. We can confirm the results are the same.

[0086] The required times for individually generating $R_{i'}$ and $R_{i''}$ through divided multi-dimensional coordinate are 0.001437 seconds and 0.001421 seconds, respectively, and the sum of those is close to the time 0.002781 seconds when undivided.

[0087] As mentioned above, even only the initial value and the multi-dimensional coordinate are changed, without changing calculation method, the subsystem is easily separated from the parent system. The calculated quantity is reduced after division, and the generation speed of the random number is faster.

[0088] In addition, because of division, it is not necessary for the lower degree system keeping the information of the upper degree system, thus, the upper degree system becomes safer from the viewpoint of information safety. Such as the situation of the product management system, when production department of certain product migrates abroad, the division of the management system is effective.

[0089] According to the present invention, the proposed method for generating and managing random numbers can generate and manage irregular M series N bits binary series R_i through one series N bits binary series R (initial value) and regular M multi-dimensional coordinate information i (FIG. 1).

[0090] According to the present invention, the random number generation is performed through R and regular i to generate R_i, in which it is characterized that it is impossible to infer another features of R_i or R through one given i or R_i. The random number generator with such features can be applied to a wide range of areas of information security.

[0091] One-time system for identification of ID and PW can be constructed through random number generation method of the present invention. That is, the secret (ID and PW) kept at system side and user side is used for R, R and new i are used for generating R_i (for identification of ID and PW) at user side, and i and R_i are delivered to system side for every identification. At system side, the received i and R of the user are used for generating R_i compared with the delivered R_i for identification. Because of one-time i, even if it is tapped, there would not be any problem. Also because of secret R without being delivered (through communication network), it is further safer. If R is made of information adopted from human body, such as fingerprint, vein, etc, it is not necessary to memorize R.

[0092] If the random number generation method of the present invention is applied to system for generating and managing key in encryption system, the so- called strongest encryption method is realized, that is, the encryption method with one-time key. Because multi-dimensional coordinate information i is only used once, it is impossible to use same key again.

[0093] If the random number generation method of the present invention is applied to common key encrypted communication, between two sides keeping same keys R, the information is encrypted by one-time key R_i. Only encrypted information and i are delivered, and it is not necessary to deliver R. The other side receiving the encrypted information and i uses kept R and i for generating R_i to recover the information. Because R is not shown in communication, it is further safer.

[0094] Because the random number generation of the present invention calculates Equation 1 through fixed decimal point calculation, the calculation can be performed through integer operations. Also because integers can easily perform division calculation, even different calculation system, such as general-purpose computer (various OS), dedicated hardware, microcomputer, etc, only basic integer arithmetic implemented, for example, addition, multiplication, bit shift, logical operation between bits, etc, same output is obtained through same input.

[0095] Therefore, the random number generation device of the present invention has various combinations adapted for "performance-cost" needed, is highly scalable, and can be applied to a wide range of industries centered on information security areas. As shown in the above, the present invention is described through embodiments, however, the technical means are not limited in the range of these embodiments. These embodiments can be modified or improved as needed. The present invention generates binary series through Equation 1, but the other methods generating binary series are available. Moreover, these embodiments can be modified on the basis of generation efficiency and features of random number. It is clear that, as for such modification, the improved patterns are included within the range of the technology of the present invention.

Claims

1. A method for generating and managing random numbers, comprising steps of:

   an input unit setting an N bits binary series R, where N is an integer and N>2, and multi-dimensional coordinate information i for K dimensions (i1, ..., ik, ..., iK) where ik is an integer and ik $\geqq$0, k:1, 2, ..., K, wherein R is a decimal portion of an initial value x_0 for a nonlinear function, $x_{t+i} = 4x_t(1 - x_t)$ (hereinafter referred to as LMAP, $0 < x_t < 1$, t=0,1,2,...), and said coordinate information i is then transformed into coordinate value of each dimension (i1, ..., ik, ..., iK) by a transforming circuit; and x_0, i1, ..., ik, ..., iK are stored in a register;

   1) according to the initial value x_0 and ik, an LMAP circuit implementing LMAP calculation having the accuracy of N bits through a fixed decimal point calculation, which is repeatedly done by chaos computing to generate N bits binary series B_{ik}, and bits of B_{ik} are individually constituted by $b_{k,0}, b_{k,1}, \ldots b_{k,N-1}$, wherein $b_{k,0} = [2 \times x_{N\times ik}]$, $b_{k,1} = [2 \times x_{N\times ik+1}]$, ...and $b_{k,N-1} = [2 \times x_{N\times ik+N-1}]$, and [] represents a calculation of

removing decimal portion;
2) a chaos computing unit implementing the chaos computing on B_{ik} as a decimal portion of the initial value x_0 after LMAP and repeatedly conducting calculation of the LMAP to generate and store an N bits binary series $R_{i1,...ik}$, where $R_{i1,...,ik}$ is R_{i1} when k=1 and $R_{i1,...,ik}$ is $R_{i1,...,ik}$ when k=K, wherein the bits of $R_{i1,...,ik}$ are individually constituted by $r_{k,0}$,$r_{k,i}$, ... and $r_{k,N-1}$, and $r_{k,0} = [2 \times x_N]$, $r_{k,1}=[2 \times x_{N+i}]$, ..., $r_{k,N-1} = [2 \times x_{2N-1}]$, which are stored in a random number register;
3) using said $R_{i1,...,ik}$ stored in the random number register as a decimal portion of initial value x_0 of the LMAP and storing it to said register, except when k=K; and
a random number generation control unit implementing the above operations 1), 2) and 3) for each dimension k=1, 2, ..., K, but when k=K, operation 3) is stopped, and operation 2) is used to generate $R_{i1,...,ik}$ as random numbers.

2. A device for generating and managing random numbers, including:

an input unit of generating and managing information for random numbers, which receives N, where N is an integer and N≧2, an N bits binary series R as an initial value information to generate random numbers, and multi-dimensional coordinates i for K dimensions (i1, ..., ik, ..., iK), where ik is an integer and ik≧0, further K is an integer and K≧1, having multi-dimensional coordinate information;
an initial value/multi-dimensional coordinates transformation unit, which transforms R, decimal portion of initial value x_0 for nonlinear function, $x_{t+1} = 4x_t(1 - x_t)$ (herein after referred to as LMAP, $0 < x_t < 1$), into x_0, and transforms I into multi-dimensional coordinates i1, ..., ik, ..., iK for the LMAP to generate random numbers;
a register, which stores the initial value x_0 and multi-dimensional coordinates i1, ..., ik, ..., iK transformed by the initial value/multi-dimensional coordinates transformation unit;
a chaos computing unit generating chaos binary series, which repeatedly implements calculation of the LMAP in N bits calculation through a fixed decimal point calculation, on the basis of the initial value x_0 and multi-dimensional coordinates stored in the register, to generate N bits binary series B_{ik}, bits of B_{ik} are individually constituted by $b_{k,0}$,$b_{k,1}$,...,$b_{k,N-1}$, wherein $b_{k,0} = [2 \times x_{N \times ik}]$, $b_{k,1} = [2 \times x_{N \times ik+1}]$, ... and $b_{k,N-1} = [2 \times x_{N \times ik+N-1}]$, and [] represents an operator used to remove decimal portion, and B_{ik} is used for the decimal portion of initial value x_0 and transformed into x_0 through the chaos computing unit repeatedly implementing the calculation of the LMAP to generate N bits binary series $R_{i1,...,ik}$ where $R_{i1,...,ik}$ is R_{i1} when k=1, and $R_{i1,...,ik}$ is $R_{i1,...,ik}$ when k=K, and the bits of $R_{i1,...,ik}$ are individually constituted by $r_{k,0}$,$r_{k,1}$,... and $r_{k,N-1}$, wherein $r_{k,0} = [2 \times x_N]$, $r_{k,1} = [2 \times x_{N+1}]$, ..., $r_{k,N-1} = [2 \times x_{2N-1}]$;
a random number register, which stores N bits binary series $R_{i1,...,ik}$ output by the chaos function computing unit; and
a random number generation control unit, which implements following operations, including an operation of the input unit of generating and managing information for random numbers, an operation of the initial value/multi-dimensional coordinates transformation unit, a calculation of the chaos function computing unit, and a calculation of x_0 transformed from $R_{i1,...,ik}$, as the decimal portion of the initial value x_0 of the LMAP, for each dimension k=1,2, ..., K, except when k=K, to generate multi-dimensional random number R_i.

Patentansprüche

1. Ein Verfahren zur Zufallszahlenerzeugung und -verwaltung, das folgende Schritte umfasst:

eine Eingabeeinheit setzt eine N Bit lange binäre Reihe R, wobei N eine ganze Zahl und N>2 ist und eine multidimensionale Koordinateninformation i für K Dimensionen (i1, ..., ik, ..., iK), wobei ik eine ganze Zahl ist und ik ≥ 0, k:1, 2, ..., K, wobei R ein Dezimalteil eines Anfangswerts x_0 für eine nichtlineare Funktion ist, $x_{t+1}=4x_t(1-x_t)$ (nachstehend bezeichnet als LMAP, $0<x_t<1$, t=0,1,2,...), und die Koordinateninformation i dann in den Koordinatenwert einer jeden Dimension (i1, ..., ik, ..., iK) durch eine Transformationsschaltung umgewandelt wird; und x_0, i1, ..., ik, ..., iK in einem Register gespeichert werden; wobei

1) gemäß dem Anfangswert x_0 und ik eine LMAP-Schaltung LMAP-Berechnung mit der Genauigkeit von N Bit durch eine festgelegte Dezimalpunktberechnung implementiert, die wiederholt durch Chaos-Berechnung erfolgt, um N Bit lange binäre Reihen B_{ik} zu erzeugen und Bit von B_{ik} einzeln durch $b_{k,0}$, $b_{k,1}$, ... $b_{k,N-1}$ dargestellt werden, wobei $b_{k,0}=[2 \times x_{N \times ik}]$, $b_{k,1}=[2 \times x_{N \times ik+1}]$, ... und $b_{k,N-1}=[2 \times x_{N \times ik+N-1}]$, und [] eine Berechnung zum Entfernen des Dezimalteils darstellt;
2) eine Chaos-Berechnungs-Einheit die Chaos-Berechnung auf B_{ik} als einen Dezimalteil des Anfangswerts

x_0 nach dem LMAP implementiert und wiederholt Berechnung des LMAP durchführt, um eine N Bit lange binäre Reihe $R_{i1, ..., tk}$ zu erzeugen und zu speichern, wobei $R_{i1, ..., ik} = R_{i1}$ ist, wenn k=1 und $R_{i1, ..., ik} = R_{i1, ..., iK}$ ist, wenn k=K, wobei die Bits von $R_{i1, ..., ik}$ durch $r_{k,0}$, $r_{k,1}$, ... und $r_{k,N-1}$, und $r_{k,0}=[2 \times x_N]$, $r_{k,1}=[2 \times x_{N+1}]$, ..., $r_{k,N-1} =[2 \times x_{2N-1}]$ einzeln dargestellt werden, die in einem Zufallszahlenregister gespeichert werden, einzeln gebildet werden;

3) die $R_{i1, ..., tk}$, die in dem Zufallszahlenregister gespeichert sind, als ein Dezimalteil des Anfangswerts x_0 des LMAP verwendet werden und diese in dem Register gespeichert werden, außer wenn k=K; und

eine Zufallszahlenerzeugunssteuereinheit die oben genannten Operationen 1), 2) und 3) für jede Dimension k=1, 2, ...,K implementiert, aber wenn k=K ist, die Operation 3) gestoppt wird und Operation 2) verwendet wird, um $R_{i1, ..., ik}$ als Zufallszahlen zu erzeugen.

2. Vorrichtung zur Zufallszahlenerzeugung und -verwaltung, mit:

einer Eingabeeinheit zum Erzeugen und Verwalten von Informationen über Zufallszahlen, die N, wobei N eine ganze Zahl ist und N≥2, eine N Bit lange binäre Reihe R als eine Anfangswertinformation, um Zufallszahlen zu erzeugen und multidimensionale Koordinaten i für K Dimensionen (i1,...,ik,...,iK) empfängt, wobei ik eine ganze Zahl und ik ≥ 0 ist, weiter K eine ganze Zahl und K≥1 ist, die multidimensionale Koordinateninformation aufweist;
einer Anfangswertmultidimensionalkoordinatentransformationseinheit, die R, Dezimalteil von Anfangswert x_0 für nichtlineare Funktion, $x_{t+1}=4x_t(1-x_t)$ (nachstehend bezeichnet als LMAP, $0<x_t<1$), in x_0 umwandelt und I in multidimensionale Koordinaten i1,...,ik,...,iK für das LMAP umwandelt, um Zufallszahlen zu erzeugen;
einem Register, das den Anfangswert x_0 und multidimensionale Koordinaten i1,...,ik,...,iK speichert, die durch die Anfangswertmultidimensionalkoordinatentransformationseinheit umgewandelt werden;
einer Chaos-Berechnung-Einheit, die chaotische binäre Reihen erzeugt, die wiederholt Berechnung des LMAPs in N Bit lange Berechnung, durch eine feste Dezimalpunktberechnung auf der Basis des Anfangswerts x_0 und multidimensionalen Koordinaten, die im Register gespeichert sind implementiert, um N Bit lange binäre Reihen B_{ik} zu erzeugen, wobei Bits von B_{ik} einzeln durch $b_{k,0}$, $b_{k,1}$, ... $b_{k,N-1}$ dargestellt werden, wobei $b_{k,0}=[2 \times x_{N\times ik}]$, $b_{k,1}=[2 \times x_{N\times ik+1}]$, ... und $b_{k,N-1}=[2 \times x_{N\times ik+N-1}]$, und [] einen Operator darstellt, der verwendet wird, um den Dezimalteil zu entfernen und B_{ik} für den Dezimalteil des Anfangswerts x_0 verwendet wird und durch die Chaos-Berechnung-Einheit in x_0 umgewandelt wird, die wiederholt die Berechnung des LMAP implementiert, um N Bit lange binäre Reihen $R_{i1,...ik}$ zu erzeugen, wobei $R_{i1, ..., ik} = R_{i1}$ ist, wenn k=1 ist und $R_{i1, ..., ik} = R_{i1, ..., iK}$ ist, wenn k=K, und die Bits von $R_{i1, ..., ik}$ einzeln durch $r_{k,0}$, $r_{k,1}$, ... und $r_{k,N-1}$ dargestellt werden, wobei $r_{k,0}=[2 \times x_N]$, $r_{k,1}=[2 \times x_{N+1}]$, ..., $r_{k,N-1}=[2 \times x_{2N-1}]$;
einem Zufallszahlenregister, der N Bit lange binäre Reihen $R_{i1, ..., tk}$ speichert, die durch die Chaosfunktionsberechnungeinheit ausgegeben werden; und
eine Zufallszahlenerzeugungssteuerungseinheit, die die folgenden Operationen implementiert, einschließlich einer Operation der Eingabeeinheit zum Erzeugen und Verwalten von Informationen über Zufallszahlen, eine Operation der Anfangswertmultidimensionalkoordinatentransformationseinheit, eine Berechnung der Chaosfunktionberechnungeinheit und eine Berechnung von x_0 umgewandelt von $R_{i1, ..., ik}$ als Dezimalteil des Anfangswerts x_0 des LMAP für jede Dimension k=1,2,...,K außer wenn k=K, um multidimensionale Zufallszahl R_i zu erzeugen.

Revendications

1. Procédé de génération et de gestion des nombres aléatoires, comprenant les étapes consistant à :

définir, par le biais d'une unité d'entrée, une série binaire R à N bits, où N est) un entier et N>2 et une information de coordonnée multidimensionnelle i pour K dimensions (i1, ..., ik, ..., iK), où ik est un entier et ik ≧0, k:1, 2, ..., K, R étant une partie décimale d'une valeur initiale x_0 pour une fonction non linéaire, $x_{t+1}=4x_t(1-x_t)$ (appelée ci-après LMAP, $0<x_t<1$, t=0,1,2,...), et ladite information de coordonnée i étant ensuite transformée en valeur de coordonnée de chaque dimension (i1, ..., ik, ..., iK) par un circuit de transformation ; et x_0, i1, ..., ik, ..., iK étant mémorisées dans un registre ;

1) en fonction de la valeur initiale x_0 et ik, implémenter, à l'aide d'un circuit LMAP, un calcul LMAP ayant la précision de N bits à travers un calcul de point décimal fixe effectué de façon répétitive par calcul par chaos pour) générer une série binaire B_{ik} de N bits et les bits de B_{ik} étant constitués individuellement par $b_{k,0}$, $b_{k,1}$, ... $b_{k,N-1}$, $b_{k,0}=[2 \times x_{N\times ik}]$, $b_{k,1}=[2 \times x_{N\times ik+1}]$, ..., et $b_{k,N-1}=[2 \times x_{N\times ik+N-1}]$, et [] représentant un calcul de retrait de partie décimale ;

2) implémenter, par le biais d'une unité de calcul de chaos, le calcul de chaos sur B_{ik} comme une partie décimale de la valeur initiale x_0 après LMAP et effectuer de façon répétitive le calcul de LMAP pour générer et mémoriser une série binaire à N bits $R_{i1, ..., tk}$, où $R_{i1, ..., tk}$ est R_{i1} lorsque k=1 et $R_{i1, ..., ik}$ est $R_{i1, ..., iK}$ lorsque k=K, les bits de $R_{i1, ..., ik}$ étant individuellement constitués par $r_{k,0}$, $r_{k,1}$, ... et $r_{k,N-1}$, et $r_{k,0}=[2 \times x_N]$,) $r_{k,1}=[2 \times x_{N+1}]$, ..., $r_{k,N-1}=[2 \times x_{2N-1}]$ mémorisés dans un registre de nombres aléatoires ;
3) utiliser ledit $R_{i1, ..., tk}$ mémorisé dans le registre de nombres aléatoires comme une partie décimale de valeur initiale x_0 de LMAP et le mémoriser dans ledit registre, sauf lorsque k=K ; et
implémenter, par le biais d'une unité de commande de génération de nombre aléatoire, les opérations 1), 2) et 3) susmentionnées pour chaque dimension k=1, 2, ..., K, mais lorsque k=K, l'opération 3) est arrêtée et l'opération 2) est utilisée pour générer $R_{i1, ..., ik}$ comme des nombres aléatoires.

2. Dispositif de génération et de gestion de nombres aléatoires, comprenant :

une unité d'entrée générant et gérant une information pour des nombres aléatoires, ladite unité recevant N, où N est un entier et N≧2, une série binaire R à N bits comme information de valeur initiale pour générer les nombres aléatoires et des coordonnées multidimensionnelles i pour K dimensions (i1, ..., ik, ..., iK), où ik est un entier et ik≧0, K étant en outre un entier et K≧1, ayant une information de coordonnée multidimensionnelle ;
une unité de transformation de valeur initiale/coordonnées multidimensionnelles qui transforme R, la partie décimale de valeur initiale x_0 pour une fonction non linéaire, $x_{t+1}=4x_t(1-x_t)$ (appelée ci-après LMAP, $0<x_t<1$), en x_0, et transforme I en coordonnées multidimensionnelles i1, ..., ik, ..., iK pour le LMAP pour générer des nombres aléatoires ;
un registre qui mémorise la valeur initiale x_0 et les coordonnées multidimensionnelles i1, ..., ik, ..., iK transformées par l'unité de transformation de valeur initiale/coordonnées multidimensionnelles ;
une unité de calcul de chaos générant une série binaire de chaos, ce qui implémente de façon répétitive le calcul de LMAP dans un calcul à N bits à travers un calcul de point décimal fixe, sur la base de la valeur initiale x_0 et des coordonnées multidimensionnelles mémorisées dans le registre, pour générer une série binaire à N bits B_{ik}, les bits de B_{ik} étant individuellement constitués par $b_{k,0}$, $b_{k,1}$, ... $b_{k,N-1}$, $b_{k,0}=[2 \times x_{N \times ik}]$, $b_{k,1}=[2 \times x_{N \times ik+1}]$, ... et $b_{k,N-1}=[2 \times x_{N \times ik+N-1}]$, et [] représentant un opérateur utilisé pour retirer la partie décimale et B_{ik} étant utilisé pour la partie décimale de valeur initiale x_0 et
transformé en x_0 à travers l'unité de calcul de chaos implémentant de façon répétitive le calcul de LMAP pour générer la série binaire à N bits $R_{i1,...ik}$, où $R_{i1,...ik}$ est R_{i1} lorsque k=1, et $R_{i1,...ik}$ est $R_{i1,...iK}$ lorsque k=K, et les bits de $R_{i1,...ik}$ étant individuellement constitués par $r_{k,0}$, $r_{k,1}$, ... et $r_{k,N-1}$, dans lequel $r_{k,0}=[2 \times x_N]$, $r_{k,1}=[2 \times x_{N+1}]$, ..., $r_{k,N-1}=[2 \times x_{2N-1}]$;
un registre de nombres aléatoires mémorisant une série binaire à N bits $R_{i1, ..., tk}$ émise par l'unité de calcul de fonction de chaos ; et
une unité de commande de génération de nombre aléatoire qui implémente les opérations suivantes, comprenant une opération de l'unité d'entrée consistant à générer et gérer l'information pour les nombres aléatoires, une opération de l'unité de transformation de valeur initiale/coordonnées multidimensionnelles,
un calcul de l'unité de calcul de la fonction de chaos et un calcul de x_0 transformé à partir de $R_{i1,...ik}$, comme la partie décimale de la valeur initiale x_0 de LMAP, pour chaque dimension k=1,2, ..., K, sauf lorsque k=K, pour générer un nombre aléatoire R_i multidimensionnel.

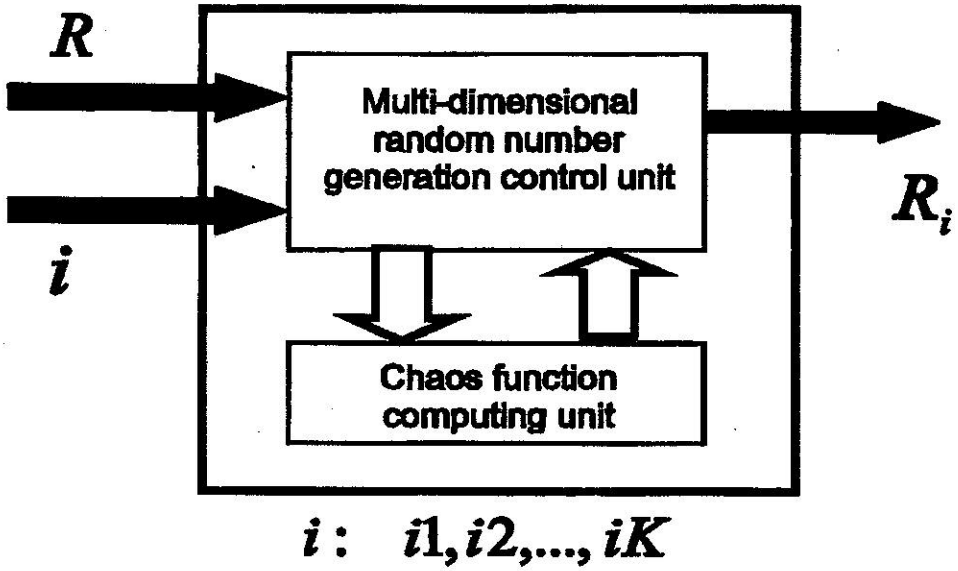
R
i
Multi-dimensional random number generation control unit
Chaos function computing unit
R_i
i : i1, i2,..., iK

FIG. 1

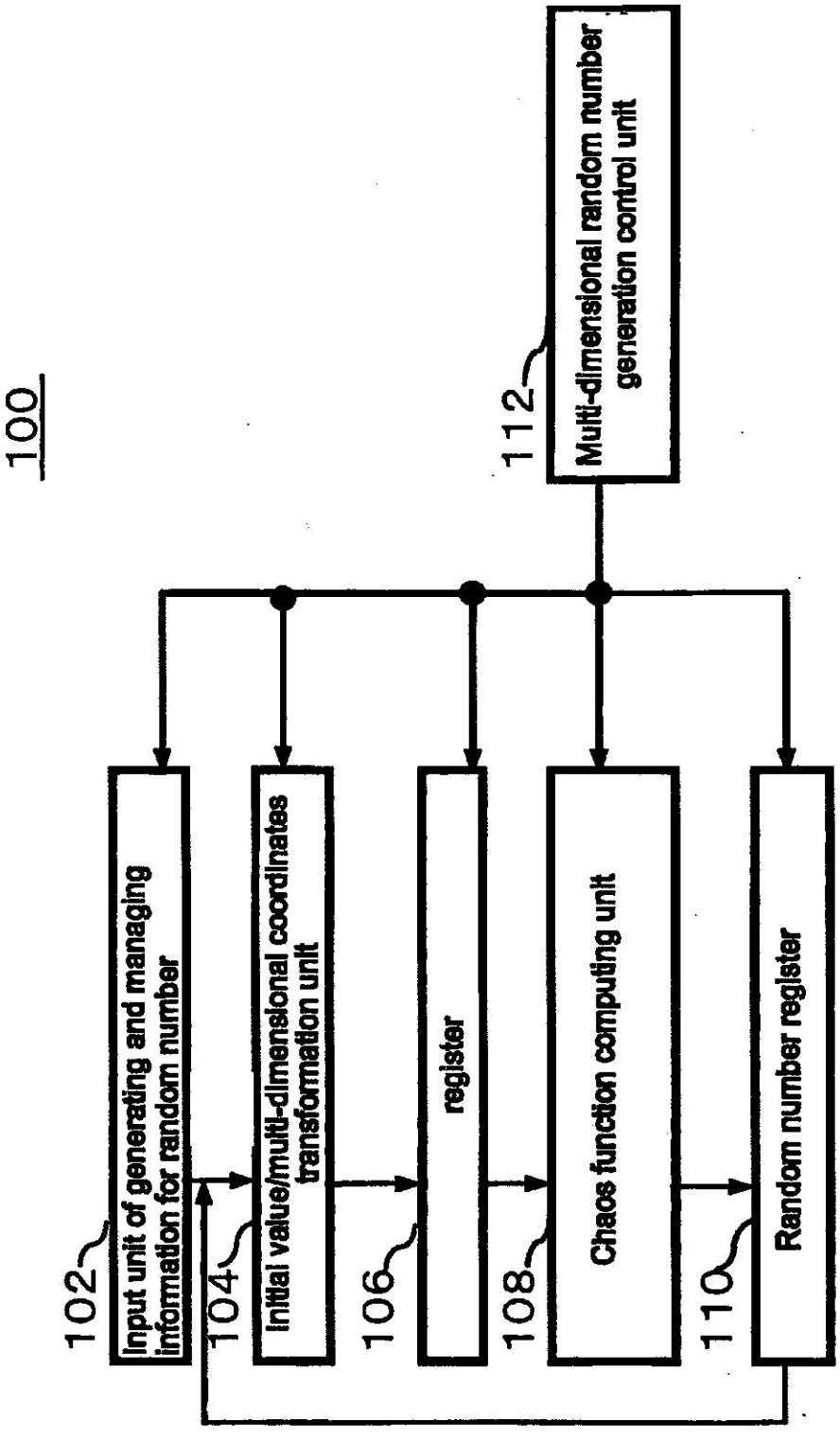
100
112
Multi-dimensional random number generation control unit
102
Input unit of generating and managing information for random number
104
Initial value/multi-dimensional coordinates transformation unit
106
register
108
Chaos function computing unit
110
Random number register

FIG. 2

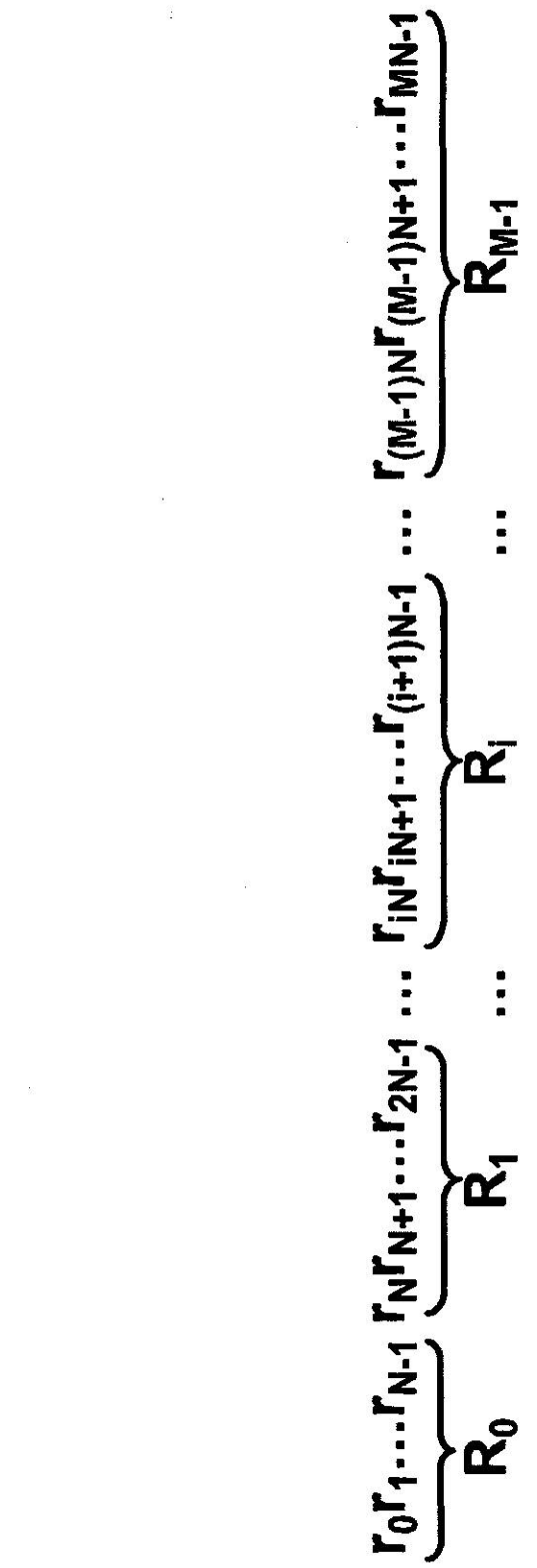
$\underbrace{r_0 r_1 \ldots r_{N-1}}_{R_0}\ \underbrace{r_N r_{N+1} \ldots r_{2N-1}}_{R_1}\ \ldots\ \underbrace{r_{iN} r_{iN+1} \ldots r_{(i+1)N-1}}_{R_i}\ \ldots\ \underbrace{r_{(M-1)N} r_{(M-1)N+1} \ldots r_{MN-1}}_{R_{M-1}}$

FIG. 3

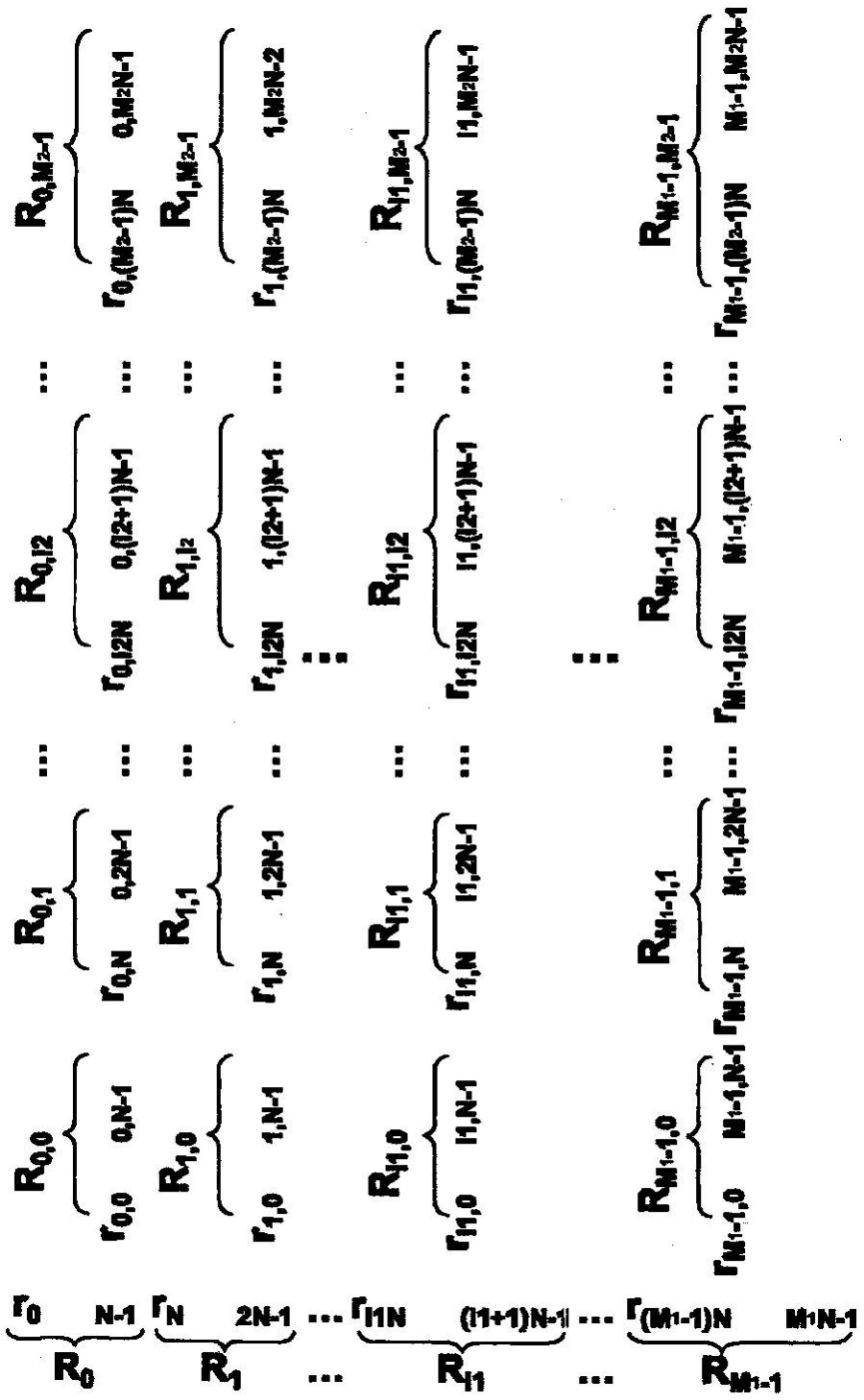

FIG. 4

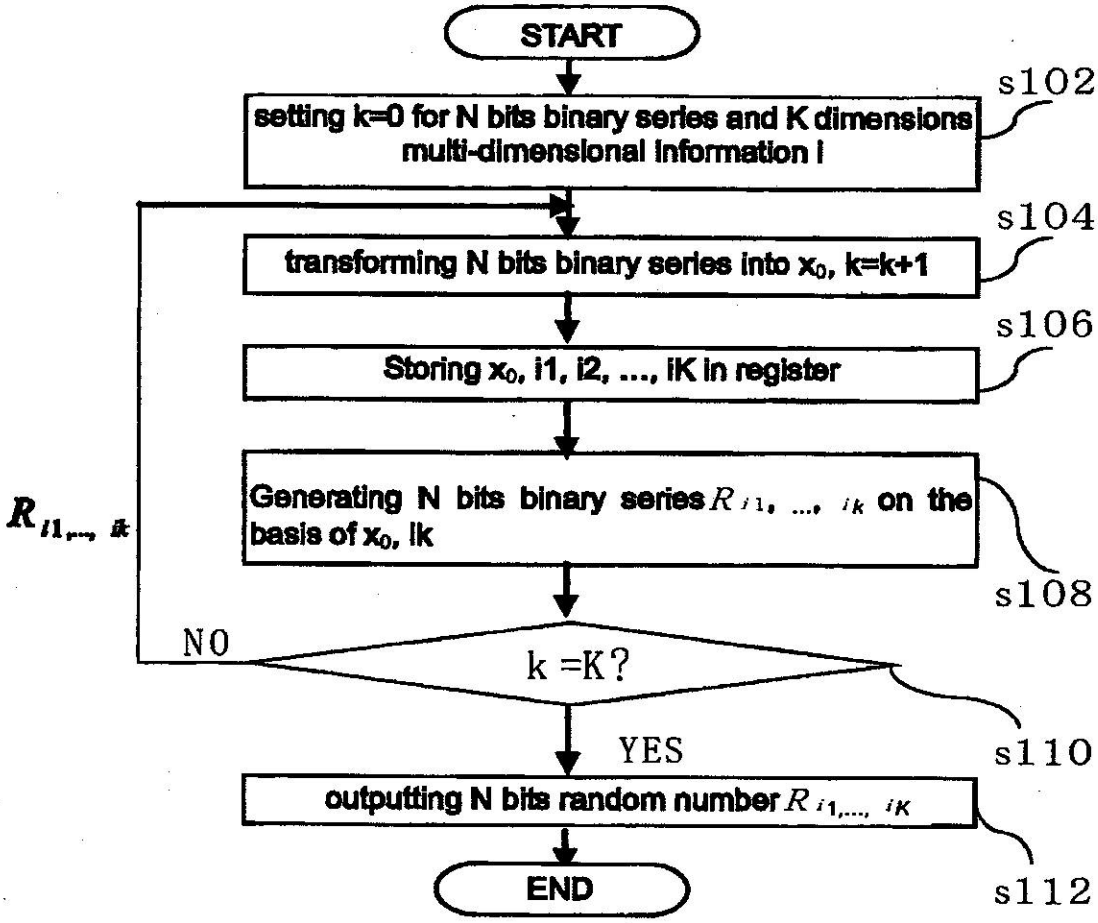
START
setting k=0 for N bits binary series and K dimensions multi-dimensional information I
s102
transforming N bits binary series into x_0, k=k+1
s104
Storing x_0, i1, i2, ..., iK in register
s106
Generating N bits binary series $R_{i1, ..., ik}$ on the basis of x_0, Ik
s108
$R_{i1,..., ik}$
NO
k =K?
s110
YES
outputting N bits random number $R_{i1,..., iK}$
s112
END

FIG. 5

	Model number							Manufacture number							
product number	M		C		780			C		42875					
hexadecimal	4D		43		2C4			43		0A77B					
ik	4	D	4	3	2	C	4	4	3	0	A	7	7	B	
k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

FIG. 6

i(i1,...i14)	t' (sec)	t (sec)
0, ···, 0(minimum)	0.000388	0.000421
4, D, 4, 3, 2, C, 4, 4, 3, 0, A, 7, 7, B	0.002441	0.002781
F, ···, F(maximum)	0.006214	0.006234

(Genuine Intel(R)CPU 1.5GHz 0.99GB RAM)

FIG. 7

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Non-patent literature cited in the description

- **ULAM, S.M. ; VON NEUMANN, J.** On Combination of Stocastic Determistic Processes. *Bull. AMS.*, 1947, vol. 53, 1120 **[0007]**
- **TOHRU KOHDA ; EIJI OGATA.** Bernoulli trials and Chaotic Trajectories in the Logistic map. *IEICE A (Japanese),* 1985, vol. J68-A (2), 146-152 **[0009]**
- **K. SHONO.** Chaos engineering. Springer-Verlag Tokyo, 2002 **[0011]**