

Números aleatorios

Historia, teoría y aplicaciones

Alfonso Manuel Mancilla Herrera*

Resumen

El desarrollo de las ciencias de la computación se ha edificado, en gran parte, sobre la base de sólidos conceptos de las matemáticas. Los números aleatorios son un buen ejemplo. Estos han sido utilizados tradicionalmente en una gran variedad de aplicaciones (juegos, criptografía, experimentos científicos, etc.), y constituyen el fundamento para el estudio y modelaje de sistemas estocásticos mediante el método de Montecarlo.

La calidad de los números aleatorios es un factor crítico de éxito para la solución de problemas en diferentes áreas; ésta es garantizada gracias a la teoría matemática y a las ventajas que ofrecen los computadores para la implementación de los diferentes métodos propuestos para la generación eficiente de dichos números.

Este artículo reseña la historia, el método para la generación de números aleatorios propuesto por D. H. Lehmer, las bases teóricas que lo soportan, las condiciones necesarias para la implementación de las rutinas generadoras en un computador y sus aplicaciones.

Palabras claves: Número aleatorio, azar, simulación, hardware, software, Montecarlo, criptografía, C.P.U.

Abstract

The development of computer sciences has been greatly built on the base of solid mathematical concepts. Random numbers are a good example. They has been traditionally used in a great variety of applications (games, cryptography, scientific experiments, etc.), and constitute the foundations for studying and modeling of stochastic systems by Montecarlo method.

The quality of random numbers is a critical success factor for solving problems in different areas. The solution is guaranteed due to the mathematical theory and the advantages the computers offer for implementing the different proposed methods for the efficient generation of these numbers.

This article reviews the history, the method by D. H. Lehmer for generating random numbers, the foundations supporting it, the necessary conditions for implementing the generating routines in a computer, and their applications.

Key Words: Random number, hazard, simulation, hardware, software, Montecarlo, Cryptography, C. P. U. O

Fecha de recepción: 10 de octubre del 2000

*Licenciado en Ciencias de la Educación, Especialidad Física y Matemática, Universidad del Atlántico, 1984. Ingeniero de Sistemas, Universidad del Norte, 1991. Diplomado en Educación Universitaria, Universidad del Norte, 1993. Especialista en Ingeniería del Software, UIS, 1995. Maestría en Ciencias Computacionales (candidato), ITESM-UNAB., 2001. (e-mail: amancill@uninorte.edu.co)

1. INTRODUCCIÓN

El hombre siempre se ha visto afectado por las fluctuaciones *aleatorias* inherentes a los fenómenos naturales que ocurren en su entorno, además ha tenido que enfrentarse, con mucha frecuencia, a la toma de decisiones bajo condiciones de *incertidumbre*; por ello, desde la antigüedad ha manifestado especial interés por temas como el azar, el futuro, el destino y se ha preocupado por hacer predicciones acerca de éstos, ya sea informalmente por medio de la adivinación, los agoreros, el azar, «la suerte», o mediante estudios formales como la futurología, la prospectiva y el estudio de las leyes del azar. Esto último, objeto de nuestro análisis, cuenta con una larga historia, por lo que inicialmente presentaremos un resumen de ésta y a continuación parte de la teoría que soporta la generación de números aleatorios para el estudio de las leyes de azar y algunas aplicaciones de dichos números en el estudio de sistemas.

2. HISTORIA

Aproximadamente por el año 3.500 a.C., juegos de azar practicados con objetos de hueso, que podrían ser considerados como los precursores de los dados, fueron ampliamente desarrollados en Egipto y otros lugares. En el siglo XVII, un noble francés, Antoine Gombauld (1607-1684), puso en tela de juicio el fundamento matemático del éxito y del fracaso en las mesas de juego. Formuló esta pregunta al matemático francés Blaise Pascal (1623-1662): «¿Cuáles son las probabilidades de que salgan dos seises por lo menos una vez en veinticuatro lanzamientos de un par de dados?» Pascal resolvió el problema, pues la teoría de la probabilidad empezaba a interesarle tanto como a Gombauld. Ambos compartieron sus ideas con el famoso matemático Pierre de Fermat (1601-1665), y las cartas escritas por los tres constituyen la primera revista académica dedicada a la probabilidad. Algunos de los problemas que ellos resolvieron habían permanecido sin solución durante unos 300 años. Sin embargo, ciertas probabilidades numéricas para ciertas combinaciones de dados ya habían sido calculadas por Giordano Cardano (1501-1576) y por Galileo Galilei (1564-1642).

Más tarde, Jacob Bernoulli (1654-1705), Abraham de Moivre (1667-1754), el reverendo Thomas Bayes (1702-1761) y Joseph Lagrange (1736-1813) inventaron fórmulas y técnicas de probabilidad. En el siglo XIX, Pierre Simon, marqués de Laplace (1749-1827), unificó esas primeras ideas y formuló la primera teoría general de la probabilidad, la cual fue aplicada inicialmente con buenos resultados a los juegos de azar; con el tiempo también se aplicó en la búsqueda de soluciones analíticas a problemas de naturaleza no determinística. La teoría de la probabilidad ha sido constantemente

N. del E.: Este artículo se presenta en una sola columna para conservar la estructura de las fórmulas.

desarrollada desde el siglo XVII y ampliamente aplicada en diversos campos de estudio. Hoy es una herramienta importante en la mayoría de las áreas de ingeniería, ciencias y administración, y se constituye en la base para el estudio de fenómenos o procesos aleatorios mediante el método de Montecarlo, que es el estudio de las leyes de azar.

En cuanto a los *números aleatorios*, podemos afirmar que la historia formal de éstos comenzó en la década de los cuarenta con el nacimiento del método llamado *Simulación de Montecarlo*, y Von Neumann, Metropolis, Ulam y Lehmer pueden ser nombrados entre los pioneros en este campo. John Von Neumann aparentemente conjeturó el potencial de los computadores para tratar problemas estocásticos en 1945 cuando escribió: «éste [el computador] ciertamente abrirá un nuevo enfoque para la estadística matemática, el enfoque para el cálculo de experimentos ...». Durante los cuarenta, la simulación de procesos estocásticos permaneció restringida al proyecto secreto del Departamento de Defensa de Estados Unidos. La publicación de *The Monte Carlo method* por N. Metropolis y Stanislaw M. Ulam en 1949 denota el inicio de la historia oficial del método. Dos años más tarde, D. H. Lehmer propuso el generador lineal de congruencia, el cual, con pequeñas modificaciones propuestas por Thomson y Rotenberg, ha llegado a convertirse en el método para la generación de números aleatorios más ampliamente usado en la actualidad. Aunque originalmente el método de Montecarlo fue implementado por John Von Neumann y Stanislaw Ulam, utilizando ruletas y dados en los problemas de difusión de los neutrones, en realidad su auge y creciente uso se debe a que hoy se emplean *números aleatorios generados por computador*.

Antes del advenimiento de las computadoras, los números aleatorios eran generados por dispositivos físicos. En 1939, Kendall y Babington-Smith publicaron 100.000 dígitos aleatorios obtenidos con un disco giratorio iluminado con una lámpara relámpago. En 1955, la *Rand Corporation* publicó un millón de dígitos producidos controlando una fuente de pulsos de frecuencia aleatoria (mecanismo electrónico); éstos se encuentran disponibles en cintas magnéticas de la *Rand*.

3. NÚMEROS ALEATORIOS

Los *números aleatorios* son aquellos que pueden ser generados a partir de fuentes de aleatoriedad, las cuales, generalmente, son de naturaleza física (dados, ruletas, mecanismos electrónicos o mecánicos), y son gobernados por las leyes del azar; éstos exhiben verdadera aleatoriedad en la realización de experimentos. Por su parte, los *números pseudo-aleatorios* son aquellos que tienen un comportamiento similar a la naturaleza aleatoria, pero están ceñidos a un patrón, generalmente de naturaleza matemática, que hace que su comportamiento sea determinístico.

Las secuencias de números aleatorios generados a partir de fuentes físicas o métodos matemáticos, basados en relaciones recursivas, deben poseer algunas propiedades:

- *Secuencias no correlacionadas*: Esto significa que en una sucesión de números aleatorios, digamos $X = \{X_0, X_1, X_2, \dots, X_{i-1}, X_i, X_{i+1}, \dots, X_{\lambda(m)-1}, X_{\lambda(m)}, \dots\}$, una subsecuencia de números aleatorios no debe estar relacionada con ninguna otra.
- *Independencia estadística y equiprobabilidad*: Lo cual implica que la probabilidad de que un número específico aparezca en la sucesión debe ser la misma para cada uno de los elementos del conjunto de números aleatorios; además, la aparición de un número dentro de la sucesión no implica ni excluye la aparición de cualquier otro.
- *Período máximo*: Los generadores de números generalmente son cíclicos, por lo cual es deseable que cada uno de los elementos del conjunto aparezca exactamente una vez en la secuencia antes de que empiecen a repetirse (ciclo completo), como se ilustra en la figura 1. El período y las propiedades de la secuencia no deben depender del valor inicial.

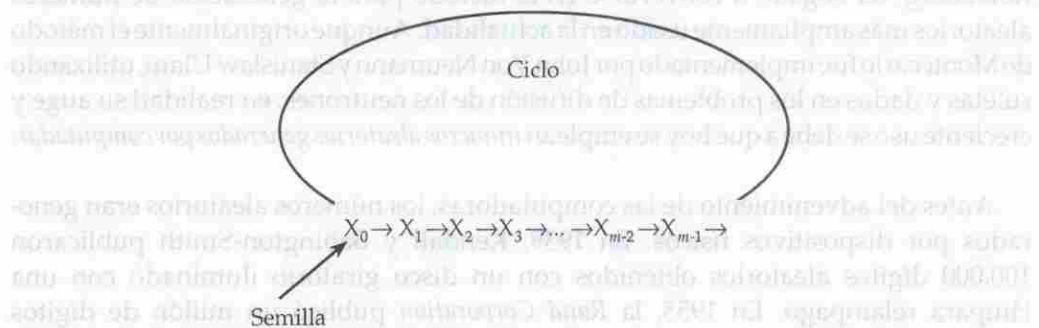


Figura 1. Ilustración de una secuencia de números aleatorios de período m .

- *Uniformidad*: Los números aleatorios pueden ser modelados mediante la función densidad de probabilidad de la variable aleatoria uniforme estándar. Esto es, si R es la fuente generadora de la sucesión de números aleatorios, entonces $R \sim U(r, a=0, b=1)$, donde $R = \frac{X}{m}$.
- *Eficiencia*: Un factor importante cuando se genera una gran cantidad de números aleatorios es la eficiencia, la cual se mide en términos de la utilización de la C.P.U. (Central Process Unit). A nivel de máquina, una propiedad importante de un generador de números aleatorios es la de producir los mismos resultados inde-

pendientemente de la plataforma computacional que se esté utilizando.

4. GENERADORES DE NÚMEROS ALEATORIOS

Los generadores de números aleatorios son procedimientos o rutinas utilizadas para generar una secuencia de números de naturaleza aleatoria. Para ello disponen de algoritmos determinísticos, que son implementados como funciones de biblioteca (Rnd, Random, Rand, Randomize, etc.), en los distintos lenguajes de programación existentes, de manera que cualquier usuario pueda utilizar éstos en sus aplicaciones o programas.

4.1. El generador Congruencial Lineal de Lehmer

La mayoría de generadores de números aleatorios que se usan actualmente son casos especiales del generador propuesto por D. H. Lehmer, en el cual se escogen 4 números mágicos del conjunto de los números enteros no negativos así:

X_0 , el valor inicial o semilla	$X_0 \geq 0$
a , el multiplicador	$a \geq 0$
c , el incremento	$c \geq 0$
m , el módulo	$m > X_0, m > a, m > c$

La secuencia deseada de números aleatorios es obtenida colocando

$$X_{j+1} = (aX_j + c) \bmod m$$

Esta es conocida como una secuencia lineal de congruencias o generador lineal de congruencias, y se denota con GLC (X_0, a, c, m). Por ejemplo, la secuencia GLC (8, 7, 5, 48) de período $\lambda(m)=12$, obtenida cuando $X_0=8$, $a=7$, $c=5$ y $m=48$, se muestra en la tabla 1.

Tabla 1
 Generador mixto GLC(8, 7, 5, 48) de periodo $\lambda^*(m) = 12$. $X_0 = X_{12} = X_{24} = \dots$
 Los números aleatorios correspondientes en se obtienen dividiendo los X_j por 48

X_0		a		c		m		$\lambda^*(m)$	
8		7		5		48		12	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	8	12	8	24	8	36	8	48	8
1	13	13	13	25	13	37	13	49	13
2	0	14	0	26	0	38	0	50	0
3	5	15	5	27	5	39	5	51	5
4	40	16	40	28	40	40	40	52	40
5	45	17	45	29	45	41	45	53	45
6	32	18	32	30	32	42	32	54	32
7	37	19	37	31	37	43	37	55	37
8	24	20	24	32	24	44	24	56	24
9	29	21	29	33	29	45	29	57	29
10	16	22	16	34	16	46	16	58	16
11	21	23	21	35	21	47	21	59	21

Analicemos detenidamente lo siguiente:

1. Los números generados corresponden siempre a el residuo de la división de (aX_j, c) entre m , por lo que los posibles valores que se obtendrán son $0, 1, 2, \dots, m-2, m-1$. Lo anterior indica que los números de la sucesión son completamente determinísticos. La aleatoriedad está asociada con el orden de aparición de éstos dentro de la secuencia. Para un valor fijo del módulo m , dicho orden depende de los tres números mágicos restantes, a saber: a, c y X_0 .
2. Al dividir los números de la sucesión X entre el módulo m obtenemos números aleatorios estandarizados en el intervalo $R = \frac{X}{m} \in [0, 1)$, los cuales, mediante métodos de transformación, como la transformada inversa, permiten la generación de valores para las variables aleatorias usadas para el modelamiento de sistemas estocásticos.
3. La semilla X_0 vuelve a aparecer luego de una serie de iteraciones, y genera un ciclo de números que se repite indefinidamente. Esta propiedad es común para todas las secuencias que tienen la forma $X_{j+1} = f(X_j)$. El ciclo repetitivo origina el período de la sucesión, el cual corresponde al número de términos diferentes generados en la secuencia, y su longitud se denota por $\lambda(m)$. En el ejemplo de la tabla 1, el período es $\lambda(m) = 12$, por lo que $X_0 = X_{12} = X_{24} = X_{36} = X_{48}$, y en general,

$$X_0 = X_{\lambda(m)} = \dots = X_{k \cdot \lambda(m)}, k \in \mathbb{Z}^+$$

- El máximo período posible se denota con $\lambda^*(m)$; éste se obtiene cuando cada uno de los potenciales residuos $[X_{j+1} = (a X_j + c) \bmod m]$ aparece exactamente una vez en la secuencia antes de que la semilla X_0 se repita.
- Algunas sucesiones se degeneran, esto es, el primer valor en repetirse en la secuencia no siempre es la semilla, esto ocurre cuando a y m no son primos relativos. En la tabla 2 presentamos un ejemplo, el generador GLC(8, 6, 5, 48). Aquí, el primer término en repetirse es X_4 , por lo que la semilla X_0 no volverá a aparecer y la sucesión se «degenera», $X_4 = X_5 = X_6$, etc.

Tabla 2
Generador mixto GLC(8, 6, 5, 48). La sucesión se degenera

$$X_4 = X_5 = X_6 = \dots$$

X_0 8		a 6		c 5		m 48		$\lambda^*(m)$ -	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	8	12	47	24	47	36	47	48	47
1	5	13	47	25	47	37	47	49	47
2	35	14	47	26	47	38	47	50	47
3	23	15	47	27	47	39	47	51	47
4	47	16	47	28	47	40	47	52	47
5	47	17	47	29	47	41	47	53	47
6	47	18	47	30	47	42	47	54	47
7	47	19	47	31	47	43	47	55	47
8	47	20	47	32	47	44	47	56	47
9	47	21	47	33	47	45	47	57	47
10	47	22	47	34	47	46	47	58	47
11	47	23	47	35	47	47	47	59	47

- Lo deseable sería obtener sucesiones de números aleatorios no «degeneradas» de período máximo. En la tabla 3 ilustramos esta situación; el generador GLC(8, 25, 5, 48) es de período máximo $\lambda^*(m) = 48$, aquí $X_0 = X_{\lambda(m)} = X_{2\lambda(m)} = \dots$. Cada uno de los potenciales residuos $\{0, 1, 2, \dots, 46, 47\}$ aparece exactamente una vez en la sucesión, antes de que la semilla (8) se repita. Observe que la aleatoriedad está asociada con el orden de aparición de los números en la sucesión y que basta cambiar la semilla para obtener otra sucesión aleatoria de período máximo, como se ilustra en la tabla 4, con el generador GLC(43, 25, 5, 48) de período $\lambda^*(m) = 48$, nuevamente $X_0 = X_{\lambda(m)} = X_{2\lambda(m)} = \dots$.

Tabla 3

Generador mixto GLC(8,25,5,48). La sucesión es de período máximo $\lambda^*(m) = 48$

$$X_0 = X_{\lambda^*(m)} = X_{2\lambda^*(m)} = \dots$$

X_0 8		a 25		c 5		m 48		$\lambda^*(m)$ 48	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	8	12	20	24	32	36	44	48	8
1	13	13	25	25	37	37	1	49	13
2	42	14	6	26	18	38	30	50	42
3	47	15	11	27	23	39	35	51	47
4	28	16	40	28	4	40	16	52	28
5	33	17	45	29	9	41	21	53	33
6	14	18	26	30	38	42	2	54	14
7	19	19	31	31	43	43	7	55	19
8	0	20	12	32	24	44	36	56	0
9	5	21	17	33	29	45	41	57	5
10	34	22	46	34	10	46	22	58	34
11	39	23	3	35	15	47	27	59	39

Tabla 4

Generador mixto GLC(43,25,5,48). La sucesión es de período máximo $\lambda^*(m) = 48$

$$X_0 = X_{\lambda^*(m)} = X_{2\lambda^*(m)} = \dots$$

X_0 43		a 25		c 5		m 48		$\lambda^*(m)$ 48	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	43	12	7	24	19	36	31	48	43
1	24	13	36	25	0	37	12	49	24
2	29	14	41	26	5	38	17	50	29
3	10	15	22	27	34	39	46	51	10
4	15	16	27	28	39	40	3	52	15
5	44	17	8	29	20	41	32	53	44
6	1	18	13	30	25	42	37	54	1
7	30	19	42	31	6	43	18	55	30
8	35	20	47	32	11	44	23	56	35
9	16	21	28	33	40	45	4	57	16
10	21	22	33	34	45	46	9	58	21
11	2	23	14	35	26	47	38	59	2

De los ejemplos anteriores se deduce que los números aleatorios no deben ser generados con métodos escogidos al azar. Alguna teoría que nos permita encontrar números mágicos que produzcan una secuencia suficientemente aleatoria de números, con un período máximo y con un mínimo de tiempo de computadora, debe ser usada.

4.2. El Método Mixto de Congruencias

Si $c \neq 0$, el generador de Lehmer también se conoce como el Método Mixto de Congruencias.

$$X_{j+1} = (aX_j + c) \bmod m$$

Su ventaja radica en su período completo que cubre el conjunto de m números diferentes cuando se tiene a m por el módulo.

4.2.1. Escogencia del módulo

El módulo puede ser cualquier entero positivo; se sugiere escoger un número grande para m , ya que éste coincide con el máximo período.

$$m = \prod_{i=1}^k p_i^{e_i}, m \in \mathbb{Z}^+, p_i \text{ es un factor primo de } m, \text{ y } e_i \geq 0$$

Para propósitos de implementación del método en un computador, m puede ser escogido de dos posibles formas:

Seleccionar m como p^e \therefore $\begin{cases} p, \text{ es la base del sistema numérico que utiliza la máquina} \\ e, \text{ el número de bits del procesador} \end{cases}$

- Seleccionar m de modo que sea el máximo número primo, menor que p^e .

4.2.2. Escogencia del multiplicador

El valor seleccionado de a debe ser entero no negativo menor que m que satisfaga:

- $(a-1)$ debe ser un múltiplo de p , para cada primo p que divide a m
- $(a-1)$ debe ser un múltiplo de 4, si m es un múltiplo de 4.

Estas dos condiciones definen el conjunto A de las a 'es

$$A = \left\{ \begin{aligned} &\{a \mid 0 \leq a < m, a \in [1]_{M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k, 4)}\} \text{ si 4 divide a } m \\ &\{a \mid 0 \leq a < m, a \in [1]_{M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k)}\} \text{ si 4 no divide a } m \end{aligned} \right.$$

Por lo que cualquier a es de la forma:

$$a = 1 + M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k, 4) * t; t \in (\mathbb{Z}^+ \cup \{0\}) \text{ si 4 divide a } m$$

$$a = 1 + M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k) * t; t \in (\mathbb{Z}^+ \cup \{0\}) \text{ si 4 no divide a } m$$

4.2.3. Escogencia de la semilla

El valor inicial o semilla (X_0) debe ser un entero no negativo menor que m .

$$X = \{x \mid 0 \leq x < m\}$$

4.2.4. Escogencia del incremento

El incremento c debe ser un entero positivo menor que m y primo relativo con m . Esta es precisamente la definición de números de Euler. La función de Euler $\Phi(m)$ nos permite calcular el número total de posibles valores para c .

$$\Phi(m) = \prod_{i=1}^k p_i^{e_i-1} (p_i - 1). \text{ Por lo tanto, hay } \Phi(m) \text{ posibles valores para } c \text{ en}$$

$$E = \{c \mid 0 < c < m, (c, m) = 1\}, (c, m) \text{ es el máximo común divisor entre } c \text{ y } m.$$

4.2.5. Ejemplo

$$\text{El módulo: } m = 48 = 2^4 * 3^1, \quad p_1 = 2, e_1 = 4 \\ p_2 = 3, e_2 = 1$$

El multiplicador: como 4 divide a 48, entonces a es de la forma

$$a = 1 + M.C.M.(p_1, p_2, 4) * t = 1 + M.C.M.(2, 3, 4) * t = 1 + 12t \rightarrow a = 1 + 12t$$

$$A = \{1, 13, 25, 37\}$$

La semilla: puede ser cualquier número del conjunto $X = \{0, 1, 2, 3, \dots, 46, 47\}$

$$\text{El incremento: hay } \Phi(48) = \prod_{i=1}^2 p_i^{e_i-1} (p_i - 1) = 2^{4-1} (2 - 1) * 3^{1-1} (3 - 1) = 16 \text{ posibles valores}$$

para c , los cuales tienen que ser primos relativos con 48, éstos son: $C = \{1, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47\}$

Con cualquier combinación de los números mágicos encontrados para $m = 48$ se obtendrán sucesiones de período máximo. En total se tienen $4 * 48 * 16 = 3.072$ sucesiones de período máximo en este caso. Los generadores GLC (8,25,5,48) y GLC (43,25,5,48), mostrados en las tablas 3 y 4, son 2 de las 3.072 potenciales sucesiones.

En general, para un m dado se tienen $n(S)$ sucesiones de período máximo, donde $n(S)$ es:

$$n(S) = \begin{cases} \frac{m}{M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k, 4)} * m * \phi(m) & \text{si 4 divide a } m \\ \frac{m}{M.C.M.(p_1, p_2, p_3, \dots, p_{k-1}, p_k)} * m * \phi(m) & \text{si 4 no divide a } m \end{cases}$$

4.3. El Método Multiplicativo de Congruencias

Si $c = 0$, el generador de Lehmer es llamado el Método Multiplicativo de Congruencias. Su principal atractivo es que las secuencias generadas presentan un mayor grado de aleatoriedad; su desventaja es que el período se reduce con relación al Método Mixto.

$$X_{j+1} = (aX_j) \bmod m$$

El máximo período que puede obtenerse es $\lambda^*(m) = M.C.M.\{\lambda(p_1^{e_1}), \lambda(p_2^{e_2}), \dots, \lambda(p_k^{e_k})\}$

$$\text{Donde } \begin{cases} \lambda(1) = 1 \\ \lambda(2^e) = \begin{cases} \Phi(2^e) = 2^{e-1} & \text{si } e = 1, 2 \\ \frac{1}{2}\Phi(2^e) = 2^{e-2} & \text{si } e \geq 3 \end{cases} \\ \lambda(p^e) = \Phi(p^e) & p, \text{ es un primo impar} \end{cases}$$

y se alcanza cuando: $(X_0, m) = 1$ y a es una raíz primitiva módulo m .

4.3.1. Escogencia del módulo

Se procede de igual forma que para el Método Mixto, esto es:

$$m = \prod_{i=1}^k p_i^{e_i}, m \in \mathbb{Z}^+, p_i \text{ es primo y } e_i \geq 0$$

Para propósitos de implementación del método en un computador, m puede ser escogido de dos posibles formas:

- Seleccionar m como p^e : $\begin{cases} p, \text{ es la base del sistema numérico que utiliza la máquina} \\ e, \text{ el número de bits del procesador} \end{cases}$
- Seleccionar m de modo que sea el máximo número primo, menor que p^e .

4.3.2. Escogencia del multiplicador

Como el multiplicador a debe ser una raíz primitiva módulo m , definiremos los conceptos de clases de congruencia módulo m y raíz primitiva.

Una clase de congruencia módulo m se denota por $[a]_m$ y define como el conjunto

$$[a]_m = \{x \mid m \text{ divide } x - a; x = a + m \cdot t; t \in \mathbb{Z}^+\}$$

El multiplicador a es una raíz primitiva módulo $m = p^e$ si y solamente si:

$$i) \quad p^e = 2^1 \rightarrow a \in [1]_2; \quad a = 1 + 2t; \quad t \in \mathbb{Z}^+$$

$$p^e = 2^2 \rightarrow a \in [3]_4; \quad a = 3 + 4t; \quad t \in \mathbb{Z}^+$$

$$p^e = 2^3 \rightarrow a \in ([3]_8 \cup [5]_8 \cup [7]_8); \quad \begin{matrix} a = 3 + 8t & t \in \mathbb{Z}^+ \\ a = 5 + 8t & t \in \mathbb{Z}^+ \\ a = 7 + 8t & t \in \mathbb{Z}^+ \end{matrix}$$

$$p^e = 2^e, e \geq 4 \rightarrow a \in ([3]_8 \cup [5]_8); \quad \begin{matrix} a = 3 + 8t & t \in \mathbb{Z}^+ \\ a = 5 + 8t & t \in \mathbb{Z}^+ \end{matrix}$$

$$ii) \quad p \text{ es un primo impar } e = 1 \rightarrow a \notin [0]_p, \quad a^{\frac{p-1}{q}} \notin [1]_p; q; \text{ factor primo de } p-1$$

$$iii) \quad p \text{ es un primo impar } e > 1 \rightarrow a \text{ debe satisfacer ii) y } a^{p-1} \notin [1]_{p^2}$$

Para determinar todos los valores de a con los que se puedan generar sucesiones de período máximo, debe descomponerse m como el producto de sus factores primos, aplicar la definición anterior y luego utilizar el teorema chino del residuo para interceptar las clases de equivalencia que resultan.

4.3.3. Escogencia de la semilla

Se escoge de la misma forma que el incremento en el Método Mixto, ya que

$(X_0, m) = 1$, por esto, hay $\phi(m) = \prod_{i=1}^k p_i^{e_i-1} (p_i - 1)$ posibles valores para X_0 en

$E = \{X_0 \mid 0 < X_0, m, (X_0, m) = 1\}$, (c, m) es el máximo común divisor entre X_0 y m

4.3.4. Ejemplo

El módulo: $m = 128 = 2^7, p_1 = 2, e_1 = 7$

El multiplicador: $p^e = 2^7, 7 \geq 4 \rightarrow a \in ([3]_8 \cup [5]_8) ; \begin{cases} a = 3 + 8t & ; t \in \mathbb{Z}^+ \\ a = 5 + 8t & ; t \in \mathbb{Z}^+ \end{cases}$

$A = \{3, 11, 19, 27, 35, \dots, 115, 123\} \cup \{5, 13, 21, 29, 37, \dots, 117, 125\}$

La semilla: hay $\Phi(128) = \prod_{i=1}^1 p_i^{e_i-1} (p_i - 1) = 2^{7-1} (2 - 1) = 2^6$ posibles valores para X_0 ,

éstos son: $X_0 = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, \dots, 125, 127\}$

Con cualquier combinación de los números mágicos encontrados para $m = 128$ se obtendrán sucesiones de período máximo. En total se tienen $64 \cdot 32 = 2.048$ sucesiones de período máximo en este caso. Los generadores GLC (125,125,0,128) y GLC (127,115,0,128) mostrados en las tablas 5 y 6, son 2 de las 2.048 potenciales sucesiones.

Tabla 5
 Generador multiplicativo GLC(125,125,0,128). La sucesión es de período máximo
 $\lambda^*(m) = 32$. $X_0 = X_{\lambda^*(m)} = X_{2\lambda^*(m)} = \dots$

X_0 125		a 125		c 0		m 128		$\lambda^*(m)$ 32	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	125	8	29	16	61	24	93	32	125
1	9	9	41	17	73	25	105	33	9
2	101	10	5	18	37	26	69	34	101
3	81	11	113	19	17	27	49	35	81
4	13	12	45	20	77	28	109	36	13
5	89	13	121	21	25	29	57	37	89
6	117	14	21	22	53	30	85	38	117
7	33	15	65	23	97	31	1	39	33

Tabla 6
 Generador multiplicativo GLC(127,115,0,128). La sucesión es de período máximo
 $\lambda^*(m) = 32$. $X_0 = X_{\lambda^*(m)} = X_{2\lambda^*(m)} = \dots$

X_0 127		a 115		c 0		m 128		$\lambda^*(m)$ 32	
j	X_j	j	X_j	j	X_j	j	X_j	j	X_j
0	127	8	95	16	63	24	31	32	127
1	13	9	45	17	77	25	109	33	13
2	87	10	55	18	23	26	119	34	87
3	21	11	53	19	85	27	117	35	21
4	111	12	79	20	47	28	15	36	111
5	93	13	125	21	29	29	61	37	93
6	71	14	39	22	7	30	103	38	71
7	101	15	5	23	37	31	69	39	101

5. OTROS GENERADORES DE NÚMEROS ALEATORIOS

5.1. Generador Shift-Register

Este generador viene dado de la forma: $X_{n+k} = (\sum_{i=0}^{k-1} a_i X_{n+i}) \pmod{2}$. En donde los X_n 's y los a_i 's son tanto 0 o 1, el máximo período viene dado por $2^k - 1$.

Existen dos formas para generar números aleatorios enteros a partir de la ecuación anterior.

La primera, conocida como el método *multi-step*, toma n sucesivos bits de la ecuación para formar un entero de n bits, entonces, n bits adicionales son generados para crear el próximo entero. El segundo, llamado el *feedback Shift-Register*, crea un nuevo entero pseudo-aleatorio para cada iteración de la ecuación. Es decir, construye el número entero con el bit generado y los $n-1$ bits generados anteriormente; por lo tanto, un nuevo número aleatorio es generado por cada bit generado.

5.2. Generador Lagged- Fibonnaci

En los últimos años se ha convertido en un generador muy popular para máquinas tanto de procesamiento serial como paralelo. Su nombre se deriva de su familiaridad con la serie de Fibonacci. Este generador intenta improvisar el generador de congruencia lineal utilizando un valor adicional para reducir las correlaciones e incrementar el período $X_n = (X_{n-j} \oplus X_{n-k}) \pmod{M}$, $j < k$, $M = 2^m$

En donde \oplus es cualquier operador aritmético binario (+, -, *, etc.). El máximo período entregado por este generador es: $(2^k - 1) \cdot 2^{m-1}$, además se pueden generar $2^{k-1} \cdot 2^{m-1}$ ciclos diferentes de período máximo.

Una de las ventajas de este generador consiste en que puede ser implementado directamente con números de punto flotante, y evitar la conversión de enteros a punto flotante que acompaña a otros generadores.

Para la implementación de este generador es necesario almacenar una tabla con los previos k números en la secuencia –es una desventajas en comparación con el Método de Congruencia Lineal–, los cuales son actualizados de manera circular a medida que el proceso es realizado.

Se ha de notar que el período puede ser incrementado a medida que el valor de k se incrementa.

5.3. Generador de Congruencia Inversa

No siendo aún ampliamente distribuido, un importante generador de números aleatorios es el Método de la Congruencia Inversa, el cual viene dado por:

$$X_n = a\bar{X}_{n-1} + b \pmod{m}$$

Donde \bar{X} denota el inverso multiplicativo, es decir, $x\bar{x} = 1$, donde $x \neq 0$ y $0\bar{0} = 0$

Eichenauer y Lehn (1986) mostraron que:

El período máximo es m . Si $X^2 - bX - a$ es un polinomio primitivo, entonces el generador devuelve una secuencia de período máximo. (Polinomio primitivo en este caso quiere decir que a y b pertenecen al conjunto de los números primos).

La escogencia de los parámetros iniciales es una de las principales ventajas del método, ya que no implica gran proceso. Una de las desventajas es que el costo de realizar el proceso del inverso modular es considerablemente alto en comparación con el Método de Congruencia Lineal.

5.4. Generador de Congruencia Lineal Combinada

Existe un método importante para la generación de números aleatorios: tomar la salida de dos generadores diferentes básicos para crear una nueva secuencia aleatoria, como sigue:

$$X_i = (A_1 \cdot X_{i-1} + C_1) \pmod{M_1}$$

$$Y_i = (A_2 \cdot Y_{i-1} + C_2) \pmod{M_2}$$

$$Z_n = X_n \oplus Y_n$$

$$Z_i = (X_i + Y_i) \pmod{\max(M_1, M_2)}$$

Donde X y Y son secuencias de dos generadores de congruencia lineal independientes.

Si el ciclo de los dos generadores independientes son números primos, implica que el ciclo del nuevo generador será la multiplicación de los períodos.

6. ALGUNAS APLICACIONES DE LOS NÚMEROS ALEATORIOS

Los números aleatorios tienen una gran cantidad de aplicaciones en diferentes áreas, entre las que podemos mencionar las siguientes:

6.1. Criptografía

En la figura 2 se ilustra la aplicación de los números aleatorios en un sistema criptográfico que utiliza *software* para seguridad de la información electrónica.

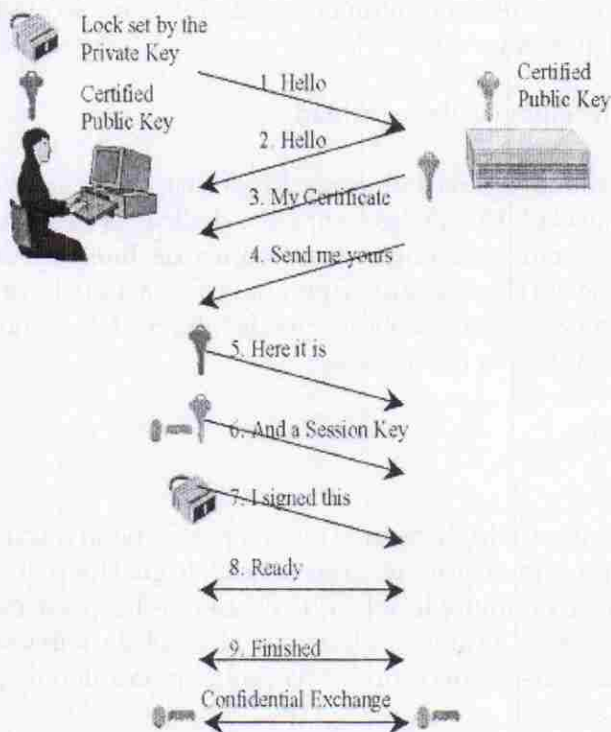


Figura 2. Aplicación de los números aleatorios en criptografía.

Imagen de <http://www.sse.ie/technology/intelrng.html>

Es claro que la seguridad de la información es importante para el comercio electrónico. Los servicios de seguridad están basados en mecanismos de criptografía, los cuales, a su vez, hacen uso de números aleatorios. Estos son utilizados para:

- Mecanismos de autenticación (usando una clave generada como número aleatorio) para proteger el mensaje.
- Mecanismos de confidencialidad utilizan llaves secretas o llaves de sesión (derivada de números aleatorios) para proteger datos durante un intercambio de información.
- Mecanismos de firma digital requieren llaves privadas, los cuales son generados de números aleatorios.

Existen compañías dedicadas al desarrollo de mecanismos de *software* para la seguridad de la información; una de ellas es *Secure Solution Experts SSE*, la cual se encuentra dedicada a proveer soluciones para sistemas de seguridad de la información electrónica a través de Internet, Intranet, e-mail, Intercambio electrónico de datos, documentos digitales, entre otros.

6.2. Hardware y tecnologías de seguridad

Sistemas de seguridad a través de *hardware* es una de las nuevas tendencias de tecnología de seguridad. Una de las compañías dedicadas a esta tecnología es *Intel*. El *Intel RNG* (Generador de números aleatorios de Intel) es un dispositivo de *hardware*, el cual mejora el proceso de criptografía, firma digital y otros protocolos de seguridad para una variedad de aplicaciones de Internet. Las grandes características que ofrece el nuevo sistema de *Intel* son:

- Alto desempeño
- Bajo costo

El *Intel RNG* utiliza ruido térmico del resistor para generar números aleatorios (no existe un algoritmo para determinar la secuencia), lo cual hace que el flujo de datos sea no-determinístico e impredecible, que es una de las principales ventajas con respecto a los números aleatorios obtenidos al nivel de *software* por algoritmos determinísticos. Este dispositivo brinda una ventaja para autenticación, integridad y seguridad de los datos.

Los tipos de aplicaciones que pueden obtener ventajas de este nuevo sistema son:

- Navegadores Web en clientes y servidores
- Redes privadas virtuales
- Correo electrónico
- Aplicaciones de comercio electrónico
- Certificados de autenticación de seguridad

6.3. Sistemas de Comunicación y Redes

Muchas universidades se encuentran encaminadas al desarrollo de diferentes proyectos en diferentes áreas de las redes y sistemas de comunicaciones; un ejemplo es la *Universidad Technische Universität Hamburg-Harburg*, en Alemania.

El proyecto hace uso de números pseudo-aleatorios en una red ATM para resolver problemas de manejo en la red, y dichos números son generados en el ámbito de *hardware* debido a la eficiencia en el proceso.

En el Departamento de Defensa de Estados Unidos existe una computadora paralelo conocida como *PENTAGON II*, la cual consta de 125 procesadores. El objetivo de la simulación es verificar y optimizar el rendimiento del sistema de cómputo de dicho Departamento que hace uso de *PENTAGON II*, esto mediante diferentes escenarios de simulación .

Otro escenario de comunicación es GPS (Sistema de Posicionamiento Global), sistema desarrollado por el Departamento de Defensa de Estados Unidos. Su principal característica es que utiliza señales de satélites codificadas, las cuales pueden ser procesadas por un receptor GPS, entonces el receptor es capaz de procesar la posición, velocidad y tiempo. Estas señales son códigos pseudo-aleatorios que se generan en los dispositivos satelitales y varían de acuerdo con la fase de comunicación.

6.4. Software para modelamiento de sistemas

En la actualidad existen herramientas que ayudan al usuario en la creación de modelos de sistemas. Entre éstos podemos mencionar:

- *Lenguajes*: SLAM, ECSL, SIMAN, los cuales son lenguajes de simulación de alto nivel, que permiten un desarrollo rápido en comparación con otros lenguajes.
- *Simuladores*: ProModel, Taylor II, los cuales son sistemas que manejan datos con poca o sin programación requerida.
- *Sistemas híbridos*: *Arena*, el cual combina la flexibilidad de un lenguaje de Simulación (SIMAN) con un sistema de manejo de datos amigable al usuario; esto permite que éste explote las características de velocidad de ambos lados.

Paradójicamente, la principal conclusión es que los números aleatorios no deben ser generados con métodos escogidos al azar. Alguna teoría matemática que nos permita encontrar una relación que produzca una secuencia suficientemente aleatoria de números, con un período máximo y con un mínimo de tiempo de computadora, debe ser usada.

Los números generados, con fundamento en la teoría estudiada y con ayuda de computadores, que logran pasar las pruebas estadísticas con respecto a su carácter aleatorio, se llaman «números pseudo-aleatorios», aunque se produzcan mediante un proceso completamente determinístico.

Existen diferentes tipos de generadores de números aleatorios (ya sea por medio de *hardware* o *software*), los cuales brindan comportamientos diferentes en la secuencia de números obtenidos de acuerdo con sus características, entre ellas, el valor inicial, eficiencia, la uniformidad, y principalmente, la longitud del período; todas ellas son importantes en el momento de analizar un generador, debido a que suministran información acerca del grado de complejidad, implementación o necesidad para los cuales son requeridos.

Como nos hemos podido dar cuenta, los números aleatorios tienen una gran importancia hoy en día para diferentes procesos, especialmente aquellos que necesitan una gran cantidad de datos numéricos para aplicaciones de simulación o aquellos en los cuales la seguridad juega un papel importante en el sistema, y para el cual los números aleatorios son una clave para lograr este objetivo.

Referencias

- LEHMER, D.H. «Mathematical Methods in Large-Scale Computing Units». In Proc. 2nd Sympos. on Large-Scale Digital Calculating Machinery, Cambridge, MA, 1949, p. 141-146, Cambridge, MA, 1951. Harvard University Press.
- KNUTH, Donald E. *The art of computer programming*. Estados Unidos, Addison-Wesley, 1969.
- LEVEQUE, William Judson. *Topics in Number Theory*. Volume I. Estados Unidos, 1965.
- LEEB, H. «Random Numbers for Computer Simulation». Master's thesis, University of Salzburg, 1995. Abstract available.
- <http://random.mat.sbg.ac.at/>
- <http://www.sse.ie>.
- <http://www.incrypt.com>.

TERMINOLOGÍA

- *Números aleatorios*: Secuencia de números generados por dispositivos físicos (dados, ruletas), electrónicos o a través del uso de relaciones matemáticas recursivas implementadas en un computador. Estos números son totalmente determinísticos, pero como pasan un conjunto de pruebas estadísticas, se les conoce como «pseudo-aleatorios o aleatorios».
- *Montecarlo*: Clave utilizada por Neumann y Ulam durante la Segunda Guerra Mundial en el proyecto «Manhattan». Este término se utiliza en muchas ocasiones como sinónimo de simulación.
- *Hardware*: Conjunto de dispositivos físicos componentes de un computador.
- *Software*: Conjunto de programas que se ejecutan en un computador.
- *Criptografía*: Ciencia que se encarga de estudiar los métodos para el cifrado y la seguridad en el manejo de la información .
- *C.P.U.* : Unidad central de procesamiento en una máquina(computador).