AUTÓMATAS CELULARES APLICADAS A LA ENCRIPTACIÓN DE DATOS

Sotelo Arellano J. M. [(1)], Ordaz Salazar F. C. [(2)]; Villarreal Zapata E. [(2)];
[(1)] Facultad de Informática
Universidad Autónoma de Querétaro
[(2)] Universidad Politécnica de San Luis Potosí

RESUMEN

Puesto que para la encriptación de datos se necesita una llave como base, es indispensable tener una muy buena y confiable, para así evitar el acceso de terceros a la información encriptada.
Para esto se requiere de un generador de números pseudo-aleatorios que nos proporcionarán dicha llave, y para ello se ha de trabajar con autómatas celulares auxiliándose de la herramienta Mathematica, para revisar qué reglas, y a qué nivel, son pseudo-aleatorias lo que nos asegura que podrá generar una llave segura.
Este proyecto se centra en la revisión de posibles reglas pseudo-aleatorias, analizando sus características detalladamente y sometiéndolas a un conjunto de pruebas de aleatoriedad con el fin de conocer cuales de ellas nos permitirán obtener los números pseudo-aleatorios que conformarán la llave para encriptación.

INTRODUCCIÓN

Ésta investigación es parte complementaria de un proyecto que se está trabajando en la Universidad Politécnica de San Luis Potosí, en el que se pretende desarrollar un sistema de encriptación de datos basado en Autómatas Celulares.
Como parte del proyecto inicial se debe comprobar que se está trabajando con reglas pseudo-aleatorias, ya que de comenzar a trabajar con reglas al azar se corre el riesgo de generar una encriptación que puede ser hackeada fácilmente, debido a la existencia de patrones repetitivos en la llave de encriptación. Por lo tanto se deben de buscar reglas de Autómatas Celulares que tengan un comportamiento aleatorio, después generar una gran cantidad de secuencias con cada regla para hacerles consecutivamente pruebas de aleatoriedad que comprueben efectivamente si su comportamiento es el esperado.

ANTECEDENTES

Los autómatas celulares fueron inventados a fines de los años cuarenta por Stanislaw Ulam y John von Neumann, quienes realizaron trabajos para crear un sistema que se replicara a sí mismo a partir de una abstracción matemática.
En 1983, Stephen Wolfran publicó algunos escritos sobre una clase de autómatas que el llamaba autómatas celulares elementales y sobre su comportamiento y las reglas que los definian. Para el 2002, Wolfram publicó su libro A New Kind of Science en el cual explica ampliamente sobre ellos, su trabajo y su importancia en todas las ramas de la ciencia.
En cuanto a la encriptación, Olu Lafe (2000) nos explica que existen un numero de patentes dadas y literatura sobre ello que incluye los trabajos de Wolfram (1985), Delahaye (1991), Guan (1987) y Gutowitz (1994).

AUTÓMATAS CELULARES

Un autómata celular, en su versión más simple, es una línea unidimensional de sitios o celdas, donde cada una es blanca o negra. El color o estado de esta celda puede cambiar conforme al tiempo. Con cada paso discreto (es decir, finito) de tiempo, las celdas se actualizan (ya sea para

mantener o cambiar su color previo) de acuerdo a la función de su estado anterior y al de las dos celdas vecinas a ella (una por el lado izquierdo y otra por el lado derecho).
También es importante saber que existen sistemas más complejos que pueden incluir más estados en las celdas, vecindarios mayores, plantillas más amplias y dimensiones adicionales, entre otros.

REGLAS

A las condiciones iniciales y de vecindad de un autómata celular se le conoce como "regla".
Existen 256 (2^8) reglas para los autómatas celulares con un estado binario variable (0,1) y una vecindad de 3. Cada una de ellas está especificada por un código decimal obtenido a partir de las 8 permutaciones para la vecindad de 3 en orden descendiente y los leemos como un código binario de 8 dígitos, lo cual nos da el número de la regla. La regla 30, por ejemplo, está definida por la configuración dada en la figura 1. Nótese que la secuencia 00011110 es la representación binaria del número 30.

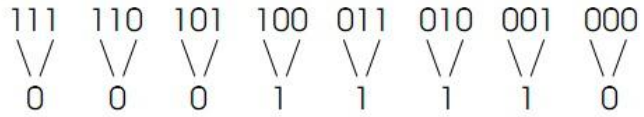

Figura 1. Representación binaria de la regla 30.

Stephen Wolfram propone un esquema de clasificación, el cual divide las reglas de Autómatas Celulares en cuatro categorías. Por tanto, tenemos 4 clases, las cuales son las siguientes: la clase 1, también conocida como de tipo fijo, la cual evoluciona rápidamente a un estado estable y homogéneo; la clase 2, también conocida como de tipo periódico, en la cual se repite un mismo patrón como un bucle; la clase 3, también conocida como de tipo caótico o pseudo-aleatorio, en donde su evolución conduce a un patrón caótico; y la clase 4, de tipo complejo, la cual presenta comportamientos tanto de la clase 2 y 3 y suelen presentar una evolución más lenta.
Existen por tanto, 38 reglas de clase 3 según menciona Stephen Wolfram, las cuales son las siguientes: 18, 22, 30, 45, 54, 60, 73, 75, 86, 89, 90, 101, 102, 105, 106, 109, 110, 120, 122, 124, 126, 129, 135, 137, 146, 147, 149, 150, 151, 153, 161, 165, 169, 182, 183, 193, 195 y 225.
Según un estudio realizado en Brazil, las reglas de clase 3 pueden ser clasificadas en 4 distintas subclases: Depósito Aleatorio (RD); Percolación Dirigida (DP); Percolación Compacta Dirigida (CDP); y Autómatas Celulares Domany-Kinzel (DKCA) y donde pueden ser simétricos o asimétricos. (Mattos & Moreira, 2004)
Siendo que las reglas de clase 3 presentan comportamientos caóticos y pseudo-aleatorios, se eligieron 4 reglas. La regla 30 (RD); la regla 54 (DKCA asimétrica); la regla 73 (CDP); y la regla 110 (DP y DKCA simétrica).

PSEUDO-ALEATORIEDAD

La necesidad de obtener números aleatorios y pseudo-aleatorios se plantea en muchas aplicaciones criptográficas, pues se emplean llaves que deben ser generadas con dichas características. El Instituto Nacional de Estándares y Tecnología (NIST) proporciona un conjunto de pruebas estadísticas de aleatoriedad y considera que estos procedimientos son útiles en la detección de desviaciones de una secuencia binaria en la aleatoriedad.
Existen dos tipos básicos de generadores usados para producir secuencias aleatorias: Generadores de Números Aleatorios (RNGs) y Generadores de Números Pseudo-Aleatorios (PRNGs). Nuestro interés está en la revisión de un generador tipo PRNGs.

El conjunto de pruebas de NIST es un paquete estadístico que consiste en 15 pruebas para probar la aleatoriedad de (arbitrariamente largas) secuencias binarias. Dichas pruebas se enfocan en diversos tipos de no aleatoriedad que pueden existir en una secuencia. Las 15 pruebas son:

1. Prueba de frecuencia (Monobit); Ésta prueba se enfoca en la proporción de ceros y unos de toda una secuencia.
2. Prueba de frecuencia dentro de un bloque; Ésta prueba se enfoca en la proporción de unos dentro de un bloque de M bits.
3. Prueba de corridas; Ésta prueba se enfoca en el número total de corridas en una secuencia, donde una corrida es una secuencia interrumpida de bits idénticos.
4. Prueba de la más larga corrida de unos en un bloque; Ésta prueba se enfoca en la corrida más larga de unos dentro de un bloque de M bits.
5. Prueba de rango de la matriz binaria; Ésta prueba se enfoca en el rango de sub-matrices disjuntas de toda la secuencia.
6. Prueba de la transformada discreta de Fourier (Espectral); Ésta prueba se enfoca en las alturas de los picos en las transformadas discretas de Fourier de las secuencias.
7. Prueba de la no acumulación de coincidencia de plantilla; Ésta prueba se enfoca en el número de ocurrencias de cadenas destino pre-especificadas. Una ventana de m bits es usada para buscar un patrón específico de m bits.
8. Prueba de acumulación de coincidencia de plantilla; Ésta prueba también se enfoca en el número de ocurrencias de cadenas destino pre-especificadas. La diferencia con la prueba anterior reside en la acción realizada al encontrar un patrón.
9. Prueba de "Estadística Universal" de Maurer; Ésta prueba se enfoca en el número de bits entre los patrones de juego (una medida que está relacionada con la longitud de una secuencia comprimida).
10. Prueba de complejidad lineal; Ésta prueba se enfoca en la longitud de un Registro de Desplazamiento con Retroalimentación Lineal (LFSR). Una baja longitud LFSR implica no aleatoriedad.
11. Prueba de serie; Ésta prueba se enfoca en la frecuencia de todos los posibles patrones de m bits acumulados a través de la secuencia completa.
12. Prueba de entropía aproximada; Ésta prueba tiene el mismo enfoque que la anterior, con el propósito de comparar la frecuencia de bloques acumulados de dos consecutivas/adyacentes longitudes (m y m+1).
13. Prueba de sumas acumulativas; Ésta prueba se enfoca en la excursión máxima (desde cero) del paseo aleatorio definido por la suma acumulada de ajustados (-1, +1) dígitos en la secuencia.
14. Prueba de excursiones aleatorias; Ésta prueba se enfoca en el número de ciclos teniendo exactamente k visitas en una suma acumulativa de un paseo aleatorio.
15. Prueba variante de excursiones aleatorias; Ésta prueba se enfoca en el número total de veces que un estado particular es visitado (es decir, se produce) en una suma acumulada de un paseo aleatorio.

METODOLOGÍA

Primeramente se recopiló información sobre las clases que propone Wolfram para clasificar las reglas del Autómata Celular. Se encontró que existían subcategorías propuestas dentro de la clase 3. Y, al encontrar estas subcategorías, se decidió realizar pruebas de aleatoriedad a una regla por división, como se mencionó anteriormente.

Por tanto, para cada una de las reglas elegidas, se generaron mediante Mathematica, 1000 archivos con 10000 datos. Estos 10000 datos son conformados a partir de una "palabra" inicial de 100 caracteres, la cual es generada aleatoriamente y se compone únicamente de 0s y 1s.
Después de generar los archivos para cada regla, estos se juntaron en un solo archivo, que posteriormente se analizaría mediante la Suite de Pruebas de la NIST. Al terminar el análisis por cada archivo final (uno por regla), se obtuvo un archivo con los resultados, lo cual nos permite ver si la regla tiene o no características que la avalen como pseudo-aleatorias. En la tabla 1, podemos ver una comparación de las reglas y su pase en cada una de las pruebas.

Tabla 1. Comparativa de los resultados de las reglas

Prueba	Regla 30	Regla 54	Regla 73	Regla 110
Frecuencia (Monobit).	**Aprobada**	Reprobada	Reprobada	Reprobada
Frecuencia dentro de un bloque	**Aprobada**	Reprobada	Reprobada	Reprobada
Corridas	Reprobada	Reprobada	Reprobada	Reprobada
Más larga corrida de unos en un bloque	**Aprobada**	Reprobada	Reprobada	Reprobada
Rango de la matriz binaria	**Aprobada**	Reprobada	**Aprobada**	Reprobada
Transformada discreta de Fourier (Espectral)	Reprobada	Reprobada	Reprobada	Reprobada
No acumulación de coincidencia de plantilla	**Aprobada**	**Aprobada**	**Aprobada**	**Aprobada**
Acumulación de coincidencia de plantilla	**Aprobada**	**Aprobada**	Reprobada	Reprobada
"Estadística Universal" de Maurer	**Aprobada**	**Aprobada**	Reprobada	Reprobada
Complejidad lineal	**Aprobada**	Reprobada	**Aprobada**	Reprobada
Serie	**Aprobada**	Reprobada	Reprobada	Reprobada
Entropía aproximada	**Aprobada**	Reprobada	Reprobada	Reprobada
Sumas acumulativas	**Aprobada**	Reprobada	Reprobada	Reprobada
Excursiones aleatorias	**Aprobada**	Reprobada	Reprobada	Reprobada
Variante de excursiones aleatorias	**Aprobada**	Reprobada	Reprobada	Reprobada

CONCLUSIONES

Como podemos ver, la regla que más propiedades de pseudo-aleatoriedad presenta es la regla 30, por lo que podemos concluir que se puede considerar que es pseudo-aleatoria. Por el contrario, la enorme falta de propiedades básicas de aleatoriedad en las otras reglas, nos permite pensar que es posible que solo las reglas de clase 3 que pertenezcan a la subcategoría RD sean las que presenten pseudo-aleatoriedad. Pero se recomienda que se realicen pruebas nuevamente, con reglas distintas a las elegidas, para comprobar si todas las reglas pertenecientes a RD presentan resultados positivos de pseudo-aleatoriedad.
Consideramos que después de realizar estas pruebas se podría continuar con el trabajo enfocándose a la encriptación y recomendamos que se pruebe cada una de las reglas pseudo-aleatorias encontradas, como llave de un sistema simple de encriptación y, posteriormente, en uno más complejo para verificar el funcionamiento de las mismas como llaves y su utilidad.

REFERENCIAS BIBLIOGRÁFICAS

- Andrew Rukhin, J. S. "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications". E.E.U.U. 2010.
- Lafe, O. "Cellular Automata Transforms". Kluwer Academic Publishers. E.E.U.U. 2000.
- Mattos & Moreira. "Universality Classes of Chaotic Cellular Automata". Belo Horizonte, MG, Brazil. 2004.
- Wikipedia. "Cellular Automaton": http://en.wikipedia.org/wiki/Cellular_automaton. 2010.