Modelización lineal de los generadores shrinking a través de las leyes 102 y 60

Sara D. Cardell
Departament d'Estadística
i Investigació Operativa
Universitat d'Alacant
Email: s.diaz@ua.es

Amparo Fúster Sabater
Instituto de Tecnologías Físicas
y de la Información
C.S.I.C.
Email: amparo@iec.csic.es

***Resumen*—En este trabajo se presenta la modelización lineal de los generadores *shrinking* y auto-*shrinking* a través de autómatas celulares lineales uniformes utilizando la ley 102 (ó la 60). La linealidad de estos autómatas se puede utilizar para el criptoanálisis de estos generadores de secuencias.**

***Palabras clave*—Autómatas celulares, criptoanálisis, generadores *shrinking*.**

I. Introducción

En la actualidad, los cifradores en flujo son los procedimientos de cifrado más rápidos que existen, por lo tanto, se utilizan en numerosas aplicaciones tecnológicas, como puede ser el algoritmo A5 para la telefonía en el GSM (véase [1]), el algoritmo E0 para Bluetooth (las especificaciones de Bluetooth se pueden ver en [2]) o el generador J3Gen para etiquetas RFID de bajo coste [3]. A través de una clave corta secreta y un algoritmo público (el generador de la secuencia), un cifrador en flujo genera una secuencia pseudoaleatoria, la secuencia cifrante. Para cifrar nuestro mensaje, éste se suma mediante operaciones XOR con la secuencia cifrante, obteniendo de este modo el texto cifrado. Para recuperar el mensaje inicial, simplemente hay que volver a ejecutar una operación XOR entre la secuencia cifrante y el texto cifrado. De aquí, se deduce la importancia de que la clave sea secreta y conocida únicamente entre las dos partes que comparten la información.

Muchos generadores de secuencias cifrantes utilizan LFSR (Linear Feedback Shift Registers) [4] de máxima longitud combinados con una función booleana no lineal. Existen otros tipos de generadores de secuencias muy populares en criptografía. Todos ellos producen secuencias de cifrado con una complejidad lineal alta, largo periodo y buenas propiedades estadísticas.

Por otro lado, se ha probado que algunos autómatas lineales de una dimensión generan exactamente las mismas secuencias que un LFSR de máxima longitud. Por lo tanto, un autómata puede ser considerado un generador alternativo a un LFSR de longitud máxima [5]. Además, algunos generadores criptográficos diseñados a través de varios LFSR pueden ser modelados como autómatas celulares lineales. En [5], [6], los autores modelizaron los generadores *shrinking* y auto-*shrinking* usando las leyes 150 y 90. La idea principal de este trabajo es modelizar esos mismos generadores utilizando las leyes 102 y 60. Estas leyes se han utilizado previamente, junto con la ley 90, en otros ámbitos, como la construcción del triángulo de Sierpinski [7], [8]. También es posible encontrar ambas leyes en el análisis de autómatas celulares complementados derivados del grupo de autómatas lineales híbridos [9] o como parte de métodos propuestos para generar familias de grafos expandidos a través de autómatas con frontera nula [10].

II. Preliminares

En esta sección, se presentan algunas definiciones necesarias para la comprensión del resto del trabajo. En la primera subsección se introducen las definiciones de generador *shrinking* y generador auto-*shrinking*. En la segunda subsección, el concepto de autómata celular es recordado.

II-A. Generadores

El **generador *shrinking*** consta de dos LFSR, R_1 y R_2, tales que la secuencia $\{a_i\}$ producida por el primer registro R_1 decima la secuencia $\{b_i\}$ producida por el segundo registro R_2. La secuencia de salida del generador $\{s_j\}$ se obtiene del siguiente modo:

$$\begin{cases} \text{Si } a_i = 1 \text{ entonces } s_j = b_i. \\ \text{Si } a_i = 0 \text{ entonces } b_i \text{ es rechazado.} \end{cases}$$

Si L_1 y L_2 son las longitudes de R_1 y R_2, respectivamente, el período de la secuencia $\{s_j\}$ es $T = (2^{L_2} - 1)2^{L_1 - 1}$, siempre que L_1 y L_2 sean primos entre sí. A su vez, la complejidad lineal, denotada por LC cumple $L_2 2^{L_1 - 2} < LC \leq L_2 2^{L_1 - 1}$. Además, es una secuencia casi equilibrada, ya que el número de unos en dicha secuencia viene dado por $2^{L_1 + L_2 - 2}$. El generador *shrinking* tiene buenas propiedades criptográficas y es fácil de implementar [11], por lo tanto, es adecuado para su implementación en sistemas de cifrado en flujo.

Por otro lado, el **generador auto-*shrinking*** fue diseñado por Meier y Staffelbach para uso en aplicaciones de cifrado en flujo [12]. Es bastante atractivo, debido a su simplicidad ya que implica el uso de un solo LFSR. Este generador consiste en un LFSR de máxima longitud que produce una secuencia que es decimada por ella misma, por lo tanto, es un caso simplificado y concreto del generador *shrinking*. La regla de decimación es bastante simple; dado un par de bits consecutivos $\{a_{2i}, a_{2i+1}\}$ de la secuencia $\{a_i\}$ generada por el LFSR, la secuencia de salida $\{s_j\}$ se obtiene del siguiente modo:

ISBN: 978-84-9717-323-0

$$\begin{cases} \text{Si } a_{2i} = 1 \text{ entonces } s_j = a_{2i+1}. \\ \text{Si } a_{2i} = 0 \text{ entonces } a_{2i+1} \text{ es rechazado.} \end{cases}$$

El periodo de la secuencia viene dado por $T = 2^{L-1}$ [12], siendo L la longitud del LFSR que genera la secuencia de entrada del generador. Además, la complejidad lineal, denotada por LC, cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$ [13].

Para ambos generadores, la clave es el estado inicial del LFSR y el polinomio de realimentación, también recomendado como parte de la clave.

II-B. Autómatas Celulares

Un **autómata celular** (CA) de una dimensión es un registro compuesto de n celdas cuyo contenido (binario en este trabajo) se actualiza de acuerdo a una ley o función de k variables [14]. Así, el estado de la celda que hay en la posición i en el instante $t+1$, x_i^{t+1}, depende del estado de las k celdas vecinas en el instante t. Si estas leyes se componen exclusivamente de operaciones XOR, entonces el autómata se dice que es **lineal**.

En un autómata, todas las celdas pueden obedecer la misma ley; en ese caso el autómata se dice que es **uniforme** o **regular**. Si obedecen distintas leyes, se dice que es **híbrido**. Por otro lado, se dice que el autómata tiene **frontera nula** o que es **nulo**, si se consideran nulos los valores adyacentes a las celdas de los extremos. Si, por el contrario, las celdas de los extremos se consideran adyacentes, se dice que el autómata tiene **frontera cíclica** o que es **periódico**.

En este trabajo, se consideran solamente autómatas uniformes de una dimensión tanto nulos como periódicos con las leyes 102 y 60. Para $k = 3$, estas leyes vienen dadas por:

Ley 102: $x_i^{t+1} = x_i^t + x_{i+1}^t$

111	110	101	100	011	010	001	000
0	1	1	0	0	1	1	0

Ley 60: $x_i^{t+1} = x_{i-1}^t + x_i^t$

111	110	101	100	011	010	001	000
0	0	1	1	1	1	0	0

Los números 01100110 y 00111100 son las representaciones binarias de 102 y 60, respectivamente. De ahí que sean llamadas ley 102 y ley 60.

La idea principal de este trabajo es proporcionar autómatas celulares lineales uniformes, donde una de las secuencias de salida corresponde a la secuencia generada por los generadores *shrinking*. La finalidad de esta modelización es expresar una secuencia pseudoaleatoria no lineal en términos de un modelo lineal, para facilitar el posible criptoanálisis de estas secuencias.

III. Modelización

Las secuencias producidas por LFSR de longitud máxima gozan de buenas propiedades criptográficas, pero sufren del *mal de la linealidad*. Gracias al algoritmo de Belekamp-Massey [15], si se intercepta como mínimo un número de bits igual al doble de la complejidad lineal de la secuencia de salida de un LSFR, ésta puede ser recuperada totalmente. Por esta razón, se introdujo el uso de funciones booleanas no lineales que rompieran esa linealidad.

Los generadores *shrinking* fueron introducidos para romper esta linealidad, sin embargo, se puede comprobar en las secciones III-A y III-B, que las secuencias producidas por estos generadores, pueden ser obtenidas a su vez como salida de una estructura lineal, autómatas celulares uniformes lineales en este caso. Esto implica que las secuencias sean sensibles a sufrir un criptoanálisis que explote esta linealidad.

III-A. Modelizando el generador shrinking

Sea $\mathbb{F}_2$ el cuerpo de Galois de dos elementos. Dados dos polinomios primitivos $p_1(x), p_2(x) \in \mathbb{F}_2[x]$, cuyos grados son L_1 y L_2, respectivamente, primos entre sí, el periodo de la secuencia de salida del generador es $T = (2^{L_2} - 1)2^{L_1 - 1}$.

Existe un autómata celular uniforme periódico de longitud T que genera la secuencia de salida utilizando la ley 102. En algunos casos, sin embargo, la longitud del autómata se ve reducida hasta llegar a ser un divisor de T. Veamos el siguiente ejemplo.

Ejemplo 1: Dados los polinomios primitivos $p_1(x) = 1+x+x^2$ y $p_2(x) = 1+x+x^3$, la secuencia generada por el generador *shrinking* tiene periodo $T = (2^3 - 1)2^{2-1} = 14$. Si se toman como estados iniciales 10 y 100, respectivamente, la secuencia de salida del generador *shrinking* es 10111000110101. En la Tabla I podemos ver un ejemplo de autómata uniforme periódico que genera la secuencia dada. Si se considera el autómata celular uniforme periódico de longitud 14 que utiliza la ley 60, en vez de la ley 102, la secuencia buscada aparecería en último lugar en vez de en primer lugar como pasa en el autómata celular de la Tabla I. De hecho, el resto de secuencias serían las mismas pero aparecerían en orden inverso. □

Diversas simulaciones para distintos valores de las longitudes L_1 y L_2 se han llevado a cabo utilizando Matlab. Para todos los casos en los que la longitud del autómata celular que genera la secuencia del generador *shrinking* es el T y no un divisor de éste, es posible observar que aparecen 2^{L_1-1} secuencias diferentes repetidas $2^{L_2} - 1$ veces y todas ellas con periodo T.

III-B. Modelizando el generador auto-shrinking

Dado un polinomio primitivo $p(x) \in \mathbb{F}_2[x]$ de grado L, sabemos que el periodo de la secuencia de salida del generador *auto-shrinking* es 2^{L-1}.

Para esta secuencia con periodo $T = 2^{L-1}$, existe un autómata celular uniforme nulo que la genera utilizando la ley 102. Observando las simulaciones obtenidas en Matlab para diversos valores de L, se deduce en todos los casos que la longitud del autómata es exactamente la complejidad lineal de la secuencia dada. Véase el siguiente ejemplo.

Tabla I
AUTÓMATA CELULAR QUE GENERA LA SECUENCIA CONSIDERADA EN EL EJEMPLO 1

102	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**
1	1	0	1	0	0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	1	0	1	0	1	1	0	0
1	0	0	1	1	0	1	1	1	1	0	1	0	0
1	0	1	0	1	1	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	0	1	0	0	1	1	0
0	0	0	1	1	1	0	1	1	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1	1	0	1
0	1	1	0	1	0	1	1	0	0	0	1	1	1
1	0	1	1	1	1	0	1	0	0	1	0	0	1
1	1	0	0	0	1	1	1	0	1	1	0	1	0
0	1	0	0	1	0	0	1	1	0	1	1	1	1
1	1	0	1	1	0	1	0	1	1	0	0	0	1
0	1	1	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	0	0	0	1	1	1	0	1	1	0

Tabla II
AUTÓMATA CELULAR QUE GENERA LA SECUENCIA CONSIDERADA EN EL EJEMPLO 2

102	**102**	**102**	**102**	**102**
0	0	0	1	1
0	0	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	1	1	1	1
0	0	0	0	1

Ejemplo 2: Dado el polinomio primitivo $p(x) = 1 + x + x^4$, la secuencia generada por el generador auto-*shrinking* tiene periodo $T = 2^3 = 8$. Si se toma el estado inicial 1001, la secuencia de salida del generador es 00011110. El polinomio característico de esta secuencia es $P_M(x) = (1 + x)^5$, por lo que la complejidad lineal será cinco. En la Tabla II se proporciona un ejemplo de autómata uniforme nulo de longitud cinco que genera la secuencia dada. Como sucedía en el ejemplo 1, si se considera el autómata celular uniforme nulo de longitud 5 que utiliza la ley 60, en vez de la ley 102, las secuencias de salida serían las mismas que en el autómata celular de la Tabla II pero aparecerían en orden inverso. □

III-C. Comparación con otros modelos lineales basados en autómatas celulares

En [5], [6], otros modelos basados en autómatas celulares híbridos y nulos fueron propuestos. En este caso, los autores propusieron autómatas celulares basados en las leyes 150 y 90, que también generaban las secuencias de salida de los generadores *shrinking*.

III-C1. Generador shrinking: En el caso del generador *shrinking*, las secuencias de salida tienen como polinomio característico $P_M(x) = P(x)^p$, con $P(x) \in \mathbb{F}_2[x]$ un polinomio de grado L_2 y p un entero tal que $2^{L_1-2} < p \leq 2^{L_1-1}$, donde L_1 y L_2 son los grados de los polinomios de realimentación de los LFSR que generan las secuencias de entrada para el generador. En [5], un algoritmo basado en la concatenación de autómatas y en el algoritmo de Cattell y Muzio [16] es propuesto. Este algorimto proporciona autómatas celulares nulos de longitud $L_2 2^{L_1-1}$ basados en las leyes 150 y 90, que generan las mismas secuencias que el generador *shrinking*.

Estos autómatas tienen una longitud notablemente inferior a la longitud de los autómatas considerados en la sección III-A. Los autómatas propuestos en este trabajo tienen una longitud igual al periodo de la secuencia, $T = (2^{L_2} - 1)2^{L_1-1}$, frente a la longitud $L_2 2^{L_1-1}$ de los autómatas considerados en [5]. Sin embargo, en los autómatas propuestos en la sección III-A aparecen 2^{L_1-1} secuencias repetidas $2^{L_2} - 1$ veces, por lo que, conociendo las primeras 2^{L_1-1} secuencias, se deduce el resto del autómata celular. Por lo tanto, tienen un comportamiento más predecible que puede ser de ayuda para recuperar la secuencia, dada una cantidad de bits interceptada. En la Tabla I, se puede observar que la primera secuencia es la misma que la tercera secuencia después de haber sufrido un traslación cíclica de cuatro posiciones. A su vez, la quinta secuencia es la misma que la segunda secuencia después de haber sufrido una traslación cíclica de cuatro posiciones también. En total, la misma secuencia aparece siete veces pero trasladada. Lo mismo ocurre con la segunda secuencia; aparece siete veces después de sufrir traslaciones de cuatro posiciones. Cuando la longitud del autómata es un divisor de T, tenemos la misma cantidad de secuencias 2^{L_1-1}, pero se repiten una cantidad de veces inferior a $2^{L_2} - 1$; hecho favorable a la hora de realizar un posible criptoanálisis.

Por otro lado, para obtener los autómatas celulares pro-

puestos en [5], había que aplicar el algoritmo de Cattell y Muzio [16] y había que aplicar, posteriormente, una concatenación. En este caso, con saber el periodo de la secuencia conocemos el autómata, ya que éste será un autómata uniforme que utilizará la ley 102 (ó la 60) en todas sus celdas, y tendrá longitud T.

III-C2. Generador auto-shrinking: Para el generador auto-*shrinking*, al ser un caso específico del generador *shrinking*, el polinomio característico es de la forma $P_M = (1+x)^p$, con $2^{L-2} < p \leq 2^{L-1}$, donde L es la longitud del LFSR que genera la secuencia de entrada para el generador. En este caso existe un autómata celular nulo de longitud 2^{L-1} basado en las leyes 150 y 90 que genera las mismas secuencias que el generador auto-*shrinking*. Los autómatas celulares obtenidos tienen una estructura definida; se considera siempre la ley 90 en las celdas de los extremos y la ley 150 en el resto de celdas (véase [6]).

Los autómatas celulares propuestos en este trabajo, poseen también una estructura definida. Para los que utilizan la ley 102, aparece siempre la secuencia de unos en último lugar y en penúltimo lugar aparece siempre una secuencia de periodo dos (la secuencia $0101\ldots$ o la secuencia $1010\ldots$). Después hay dos secuencias de periodo cuatro, cuatro secuencias de periodo ocho y, así sucesivamente, hasta encontrar 2^{L-3} secuencias de periodo 2^{L-2}. El resto de secuencias (la longitud del autómata menos 2^{L-2}) son de periodo 2^{L-1}, incluida la secuencia de salida del generador auto-*shrinking*. Por otro lado, sabemos que la complejidad lineal cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$, por lo tanto, la longitud de estos autómatas (exactamente LC) es menor que 2^{L-1}, la longitud de los autómatas propuestos en [6], con lo que se reduce la complejidad de computar estos modelos.

En la Tabla II, podemos observar que de las cinco secuencias, existe una secuencia de periodo uno, la secuencia de unos, en el último lugar. También aparece la secuencia de periodo dos en el penúltimo lugar y otras dos secuencias de periodo cuatro. Por último aparece una sola secuencia de periodo ocho, la secuencia de salida del generador auto-*shrinking*. Para modelizar esta misma secuencia utilizando las leyes 90 y 150, se necesita un autómata celular de longitud seis (véase [6]).

IV. Aplicaciones

La principal aplicación de modelizar las secuencias de salida de este tipo de generadores es el criptoanálisis, ya que estos generadores tienen buenas propiedades criptográficas y son adecuados para ser usados en cifrado en flujo. Dado un modelo lineal que describe el comportamiento del generador, el criptoanálisis puede llevarse a cabo utilizando diferentes herramientas.

- En primer lugar, se puede hacer una búsqueda exhaustiva entre todos los estados iniciales posibles del autómata celular. En el caso del generador *shrinking*, si los grados de los dos polinomios que se utilizan para generar las secuencias de entrada son L_1 y L_2, respectivamente, el autómata tendría longitud $(2^{L_2}-1)2^{L_1-1}$. La complejidad computacional de probar todos los estados sería proporcional a $2^{(2^{L_2}-1)2^{L_1-1}}$. Sin embargo, en este autómata aparecen 2^{L_1-1} secuencias repetidas $2^{L_2}-1$ veces, por lo que sólo sería necesario generar las 2^{L_1-1} primeras secuencias. Una vez generadas estas secuencias, se puede deducir el resto de las secuencias. La complejidad ascendería a $2^{2^{L_1-1}}$. En este caso, sería más eficiente hacer una búsqueda exhaustiva entre los posibles estados iniciales de los LFSR, ya que habría que considerar $2^{L_1+L_2}$ casos. Para el generador auto-*shrinking*, sucede algo parecido. Sea L la longitud del LFSR que genera la secuencia de entrada para este generador. Si se quisieran probar todos los estados iniciales del LFSR, el número de estados asciende a 2^L. Por otro lado, la longitud del autómata LC, cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$. Por lo tanto, el número total de estados iniciales del autómata que se tendrían que probar 2^{LC}, es mayor que $2^{2^{L-2}}$ y, a su vez, $2^{2^{L-2}} > 2^L$ para $L > 4$.
- Por otro lado, si se recupera una cantidad de bits consecutivos igual a $2^{L_1-1}+t$, siento t la longitud de la traslación cíclica que sufren las secuencias obtenidas por los autómatas que generan la secuencia de salida del generador *shrinking*, podemos recuperar la secuencia entera. Es posible ver que esta traslación se puede obtener dividiendo la longitud del autómata entre el número de secuencias diferentes, esto es, 2^{L_1-1}. Por lo tanto, con 2^{L_1} bits consecutivos es posible recuperar la totalidad de la secuencia. Nótese que esta cantidad es menor que la complejidad lineal de la secuencia, véase la sección II-A. Para el generador auto-*shrinking*, basta interceptar una cantidad de bits igual a la complejidad lineal de la secuencia para recuperar la totalidad de ésta. En ambos casos cabe resaltar que la complejidad lineal es la mitad de la cantidad de bits que se necesitan para llevar a cabo el algoritmo de Berlekamp-Massey [15].
- Es posible combinar los autómatas celulares propuestos en este trabajo y los autómatas celulares propuestos en [5], [6] con les leyes 150 90, para recuperar pequeñas cantidades de bits de las secuencias del autómata, que pueden ayudar a recuperar la totalidad de la secuencia. Por ejemplo, cuando el autómata que modeliza el generador *shrinking* comienza con la ley 150, las dos primeras secuencias de estos autómatas y los propuestos en la sección II-A, son las mismas. Por lo que recuperar una parte de una de las secuencias en uno de los autómatas nos lleva a recuperar otra parte en el otro autómata.

V. Conclusión

Los esfuerzos por parte de los criptógrafos de incluir generadores por decimación con la finalidad de romper la linealidad de las secuencias generadas por LFSR han sido inútiles, ya que, las secuencias de salida de estos generadores pueden modelizarse como secuencias de salida de estructuras lineales. Este trabajo analiza una familia de autómatas celulares uniformes lineales, basados en la ley 102 (60) que describen el comportamiento de los generadores *shrinking*, diseñados como no lineales.

AGRADECIMIENTOS

El trabajo del primer autor ha sido financiado por una beca postdoctoral de la Generalitat Valenciana con referencia APOSTD/2013/081 y por el proyecto MTM2011-24858 del Ministerio de Ciencia e Innovación del Gobierno de España.

El trabajo del segundo autor ha sido financiado por el Ministerio de Ciencia e Innovación del Gobierno de España bajo el proyecto "TUERI: Technologies for secure and efficient wireless networks within the Internet of Things with applications to transport and logistics", TIN2011-25452.

REFERENCIAS

[1] GSM, Global Systems for Mobile Communications, http://cryptome.org/gsm-a512.htm

[2] Bluetooth, Specifications of the Bluetooth system, http://www.bluetooth.com

[3] A. Peinado, J. Munilla, y A. Fúster-Sabater, "EPCGen2 Pseudorandom Number Generators: Analysis of J3Gen," *Sensors*, vol. 14, no. 4, pp. 6500–6515, 2014.

[4] S. W. Golomb, *Shift Register-Sequences*. Laguna Hill, California: Aegean Park Press, 1982.

[5] A. Fúster-Sabater y P. Caballero-Gil, "Linear solutions for cryptographic nonlinear sequence generators," *Physics Letters A*, vol. 369, pp. 432–437, 2007.

[6] A. Fúster-Sabater, M. E. Pazo-Robles, y P. Caballero-Gil, "A simple linearization of the self-shrinking generator by means of cellular automata," *Neural Networks*, vol. 23, no. 3, pp. 461–464, 2010.

[7] S. Wolfram, "Computation theory of cellular automata," *Communications in Mathematical Physics*, vol. 96, no. 1, pp. 15–57, 1984.

[8] S. Wolfram, *A new kind of science*. Wolfram-Media, 2002.

[9] S. Cho, U. Choi, H. Kim y Y. Hwang, "Analysis of complemented CA derived from linear hybrid group CA," *Computers and Mathematics with Applications*, vol. 53, no. 1, pp. 54–63, 2007.

[10] S. Cho, U. Choi, H. Kim, Y. Hwang y J. Kim, "60/102 Null Boundary Cellular Automata based expander graphs," in *16th Intl. Workshop on CA and DCS*. Discrete Mathematics and Theoretical Computer Science (DMTCS) Proceedings, 2010, pp. 19–28.

[11] D. Coppersmith, H. Krawczyk, y Y. Mansour, "The shrinking generator," in *Advances in Cryptology – CRYPTO '93*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1993, vol. 773, pp. 23–39.

[12] W. Meier y O. Staffelbach, "The self-shrinking generator," in *Advances in Cryptology – EUROCRYPT 1994*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1994, vol. 950, pp. 205–214.

[13] S. R. Blackburn, "The linear complexity of the self-shrinking generator," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2073–2077, 1999.

[14] A. K. Das, A. Ganguly, A. Dasgupta, S. Bhawmik, y P. P. Chaudhuri, "Efficient characterisation of cellular automata," *IEE Proceedings E: Computers and Digital Techniques*, vol. 137, no. 1, pp. 81–87, 1990.

[15] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[16] K. Cattell y J. C. Muzio, "One-dimensional linear hybrid cellular automata," *IEEE Transactions on Computer-Aided Design*, vol. 15, no. 3, pp. 325–335, 1996.