Diseño de cifradores en flujo DLFSR con alta complejidad lineal para implementación hardware

A. Peinado
Universidad de Málaga
Andalucía Tech
ETSI Telecomunicación
Depto. Ingeniería de Comunicaciones
Email: apeinado@ic.uma.es

J. Munilla
Universidad de Málaga
Andalucía Tech
ETSI Telecomunicación
Depto. Ingeniería de Comunicaciones
Email: munilla@ic.uma.es

A. Fúster Sabater
Instituto de Tecnologías
Físicas y de la Información (ITEFI)
Consejo Superior de Investigaciones
Científicas (C.S.I.C.), Madrid
Email: amparo@iec.csic.es

***Resumen*—Muchos generadores de secuencias pseudoaleatorias de uso criptográfico se basan en registros de desplazamiento con realimentación dinámica (DLFSR) para incrementar el período y la complejidad lineal de las secuencias *PN*. En este trabajo se presenta un modelo teórico que permite el diseño de secuencias más largas y con mayor complejidad lineal que las obtenidas en otros esquemas de DLFSR. El modelo determina asimismo la relación constante entre período y complejidad lineal para estas estructuras. Las secuencias aquí obtenidas presentan mejores parámetros criptográficos que las de otras propuestas de registros de desplazamiento con realimentación dinámica encontradas en la literatura.**

***Palabras clave*—cifrado en flujo (*stream cipher*), complejidad lineal (*linear span*), generador de números pseudoaleatorios (*PRNG*), realimentación dinámica (*dynamic feedback*), registro de desplazamiento realimentado linealmente (*LFSR*), secuencia binaria (*binary sequence*).**

I. Introducción

Los registros de desplazamiento con realimentación lineal (LFSRs) se han utilizado tradicionalmente como bloques básicos para la implementación de generadores de secuencia con fines criptográficos [6]. Sus secuencias de salida, las *PN* secuencias, presentan buenas propiedades de pseudoaleatoriedad (equilibrio entre ceros y unos, excelente distribución de rachas, buena autocorrelación, etc.) pero son fácilmente previsibles debido a la linealidad inherente a estas estructuras. Con el fin de romper dicha linealidad, pero a la vez manteniendo las características de pseudoaleatoriedad, se aplican diferentes técnicas de diseño como son el filtrado no lineal, la decimación irregular de *PN* secuencias o la introducción de elementos típicos de los cifradores en bloque (cajas de sustitución, vueltas de generadores en bloque conocidos, funciones de expansión de claves, etc.).

Otra técnica general para romper la linealidad de los LFSRs consiste en la modificación dinámica de los parámetros de realimentación. Entre los diferentes ejemplos de aplicación de esta técnica pueden enumerarse los siguientes:

En 2008 Che *et al.* [3] propusieron una modificación del estado del LFSR para diseñar un generador de números aleatorios. Sin embargo, en 2011 este esquema fue rechazado cuando Meliá-Seguí *et al.* [11] detectaron ciertas debilidades que cuestionaban la aleatoriedad de la secuencia de salida. A su vez, Hellebrad [8] y Rosinger [16] propusieron sendos generadores de secuencia para testeo de circuitos basados en modificaciones dinámicas de las semillas (estados iniciales) y de los polinomios de realimentación de los LFSRs.

En 2002 Mita *et al.* [12] diseñaron un generador de secuencia pseudoaleatoria basado en un LFSR con realimentación dinámica cuyo polinomio de realimentación se actualizaba según fuera el estado de otro LFSR secundario. Esta estructura puede considerarse como el inicio de los generadores DLFSR (Dynamical LFSR). Posteriormente en 2005, Babbage *et al.* diseñaron el cifrador en flujo Mickey [1] compuesto por dos LFSRs conectados entre sí de manera que cada uno de ellos controlaba la realimentación del otro. Sin embargo en 2003 Ding *et al.* [5] criptoanalizaron dicho generador.

En 2007 Kiyomoto *et al.* [9] propusieron el cifrador K2, basado en dos LFSRs y un filtro no lineal. En dicho cifrador, un bit del estado del LFSR secundario controlaba la realimentación del LFSR principal. Posteriormente, Bogdanov *et al.* [2] presentaron una evaluación positiva de la seguridad del cifrador K2.

El generador Rakaposhi fue propuesto en 2009 por Cid *et al.* [4]. Se compone de un LFSR cuyo polinomio de realimentación se selecciona entre 4 posibles opciones codificadas por 2 bits del estado de un registro con realimentación no lineal (NLFSR). La secuencia de salida se obtiene aplicando un filtro no lineal a ambos registros (LFSR y NLFSR). Recientemente, en 2013 Orumiehchiha *et al.* [13] han detectado ciertas vulnerabilidades en dicho cifrador.

En 2013, se presentó un generador de números aleatorios, el J3Gen [10], que utilizaba un LFSR cuyo polinomio de realimentación se seleccionaba de una lista de polinomios mediante un esquema cíclico. También en 2013 Peinado *et al.* [14] desarrollaron un modelo matemático, basado en secuencias entrelazadas [7], para calcular el período y la complejidad lineal de los generadores DLFSR. Posteriormente, dicho modelo se aplicó al generador pseudoaleatorio descrito en [12].

En este trabajo, se presenta una extensión del modelo matemático para DLFSRs que permite generar secuencias con mayor período y complejidad lineal que aquellas obtenidas en las referencias anteriores. Los resultados aquí descritos

ISBN: 978-84-9717-323-0

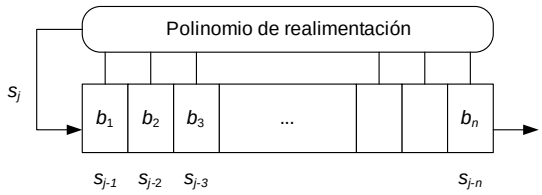

Figura 1. Estructura de un LFSR

mejoran la seguridad y robustez de las propuestas basadas en módulos DLFSR.

II. Registros de desplazamiento con realimentación dinámica (DLFSR)

Un LFSR es un registro de desplazamiento compuesto por n celdas de memoria con contenido binario $b_1, b_2, \ldots, b_n$ que funcionan de forma síncrona. A cada golpe de reloj, el contenido de cada celda se desplaza una posición a la derecha según se observa en Fig. 1. Mediante una función de realimentación representada por el polinomio de realimentación se genera un nuevo contenido para la celda b_1. En este trabajo se considerarán únicamente celdas con contenido binario, sin embargo también se han diseñado registros de desplazamiento cuyas celdas contienen elementos en un cuerpo de Galois extendido $GF(2^n)$. Si el bit de salida de la función de realimentación en el instante j es s_j, entonces el estado del LFSR de n celdas que produce s_j es $(s_{j-1}, s_{j-2}, \ldots, s_{j-n})$. Por tanto,

$$s_j = c_1 s_{j-1} + c_2 s_{j-2}, \ldots, c_n s_{j-n}, \qquad (1)$$

donde $c_1, \ldots, c_n$ son los coeficientes binarios del polinomio de realimentación

$$p(x) = c_n x^n + c_{n-1} x^{n-1} + \ldots + c_2 x^2 + c_1 x + 1. \qquad (2)$$

Es bien conocido [6] que el máximo período de la secuencia de salida de un LFSR es $2^n - 1$, es decir el registro pasa por todos los posibles estados distintos de cero. Esto sucede cuando el polinomio de realimentación es un polinomio primitivo, en cuyo caso la secuencia generada por el LFSR es una *PN* secuencia o secuencia de longitud máxima. Dichas secuencias presentan un perfecto equilibrio y distribución estadística de ceros y unos a la vez que una autocorrelación bivaluada. Es decir satisfacen perfectamente los postulados de pseudoaleatoriedad de Golomb [6]. Sin embargo el conocimiento de solamente $2n$ bits de dicha secuencia permite reconstruirla en su totalidad, ya que los coeficientes del polinomio de realimentación pueden obtenerse como solución de un sistema de n ecuaciones lineales.

Un DLFSR es un tipo de LFSR en el que el polinomio de realimentación va cambiando a medida que el registro se va desplazando. Tal y como se muestra en la Fig. 2, el modelo conceptual de un DLFSR consiste en un LFSR principal más un módulo adicional que controla el instante en el que se aplica

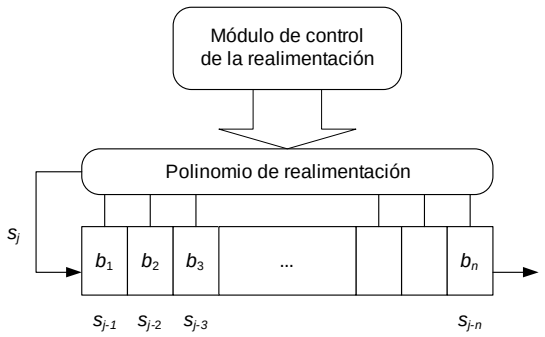

Figura 2. Estructura de un DLFSR

un nuevo polinomio de realimentación. Las secuencias generadas por un DLFSR pueden considerarse como la concatenación de segmentos de *PN* secuencias, de tal forma que el estado final del LFSR correspondiente al polinomio de realimentación $p_i(x)$ coincide con el estado inicial del LFSR correspondiente al polinomio de realimentación $p_{i+1}(x)$.

La finalidad de un DLFSR es la de generar secuencias con un período más largo y una complejidad lineal mayor que las producidas por el simple LFSR. Para llevar a cabo esta tarea, el modulo de control modifica diferentes parámetros de realimentación generando así una secuencia distinta. Los parámetros de realimentación de un DLFSR pueden enumerarse tal y como sigue:

- n : Longitud del LFSR a la vez que el grado del polinomio de realimentación.
- N_p : Número de diferentes polinomios de realimentación que van a aplicarse sobre el LFSR. En general estos polinomios son primitivos aunque también existen esquemas que incluyen polinomios no primitivos. La mayoría de diseños de DLFSR seleccionan polinomios con muchos coeficientes comunes para simplificar así la implementación hardware.
- e_i : Número de bits consecutivos generados por el mismo polinomio $p_i(x)$. Este parámetro puede ser fijo o variable.
- *Regla de selección*: Determina el orden en el que se aplican los distintos polinomios de realimentación. En algunos casos se aplican siguiendo un orden predeterminado; en otros, de forma completamente aleatoria.
- *Módulo de control*: El módulo de control establece el modo en el que se establece el instante en el que se cambia el polinomio de realimentación, así como el polinomio a utilizar o el número de bits que se van a generar. Este modo puede ser dependiente o independiente del LFSR principal, empleando señales externas al LFSR principal, como por ejemplo, LSFR secundarios; o utilizando el contenido de determinadas celdas del propio LFSR principal. Los mayores valores de complejidad lineal se alcanzan cuando se utilizan dispositivos adicionales, y por tanto, esquemas de control independientes.

Como ejemplo ilustrativo analizamos el DLFSR propuesto por Mita *et al.* en [12] que consta de un LFSR principal con $n = 16$ celdas y $N_p = 4$ polinomios de grado 16. La

regla de selección utilizada establece que los N_p polinomios se aplican siempre en el mismo orden, generanado cada uno de ellos siempre el mismo número de bits. El modulo de control está compuesto por un LFSR secundario de $m = 5$ celdas y polinomio de realimentación primitivo de grado 5. El LFSR secundario se conecta a un decodificador que, de acuerdo con su estado actual y una regla de selección fija, toma de una tabla el correspondiente polinomio a aplicar sobre el LFSR principal. A cada polinomio se le asigna un único estado del LFSR secundario. Cuando este LFSR alcanza dicho estado, el polinomio $p_i(x)$ actua sobre el LFSR principal. Por tanto sólo 4 estados del secundario modifican la realimentación del registro principal. Los 4 polinomios se aplican siguiendo una rotación cíclica. Sumando el número de bits consecutivos e_1, e_2, e_3, e_4 generados respectivamente por $p_1(x), p_2(x), p_3(x), p_4(x)$, se obtiene el período completo de la secuencia producido por el LFSR secundario, esto es

$$\sum_{i=1}^{4} e_i = 2^5 - 1 = 31. \tag{3}$$

Por otra parte, el cifrador en flujo Rakaposhi [13] es un DLFSR compuesto por un LFSR principal de longitud $n = 192$ bits con $N_p = 4$ polinomios de realimentación de grado 192. El módulo de control es un registro de desplazamiento con realimentación no lineal (NLFSR) de 128 celdas, dos de las cuales se emplean para seleccionar el polinomio de realimentación del LFSR principal. Por tanto, e_1, e_2, e_3, e_4 se determinan dinámicamente mediante los valores que toman 2 bits del NLFSR.

III. Período y complejidad lineal de las secuencias generadas

Desde un punto de vista criptográfico, el período y la complejidad lineal son dos indicadores fundamentales de la pseudoaleatoriedad de una secuencia. Ambas propiedades se definen tal y como sigue.

Definición 1. Sea $s = (s_0, s_1, s_2, \ldots) = (s(t))\ t \geq 0$ una secuencia binaria. Si existe un entero $r > 0$ tal que $s(t) = s(t + r)$ para todo $t \geq 0$, entonces la secuencia s se dice periódica y su período, representado por $T(s)$, es r.

Definición 2. La complejidad lineal de una secuencia s, representada por LC, es la longitud del LFSR más corto que puede generar dicha secuencia.

Para determinar el período y la complejidad lineal de las secuencias producidas por DLFSRs, hay que tener en cuenta que dichas secuencias son *secuencias entrelazadas* en el sentido dado en [7]. Es decir que la secuencia de salida de un DLFSR puede descomponerse en diferentes secuencias decimadas, todas ellas generadas por el mismo polinomio de realimentación. Una secuencia decimada $w_j(t)$ se construye tomando uno de cada N_s bits de la secuencia $s(t)$ empezando en $s(j)$, es decir $w_j(t) = s(j + tN_s)\ t \geq 0$. Este hecho ya fue señalado en [14] dando lugar a un modelo matemático para la generación de números pseudoaleatorios mediante DLFSRs, que puede resumirse en la siguiente ecuación:

$$M = \prod_{t}^{t+N_s} A_t, \tag{4}$$

donde A_t es una matriz $n \times n$ cuyo polinomio característico es el polinomio $p_t(x)$ aplicado al DLFSR en el instante t. El parámetro N_s es el período de aplicación de los distintos polinomios de realimentación a la vez que coincide con el número de secuencias decimadas que constituyen la secuencia entrelazada. Como ejemplo ilustrativo, podemos decir que el DLFSR definido en [12] utiliza 4 polinomios $p_1(x)$, $p_2(x)$, $p_3(x)$ y $p_4(x)$ de la siguiente manera: $p_1(x)$ genera 9 bits consecutivos, $p_2(x)$ 5 bits, $p_3(x)$ un único bit y $p_4(x)$ genera 16 bits consecutivos de su correspondiente *PN* secuencia. A continuación $p_1(x)$ generaría de nuevo 9 bits y así sucesivamente. Por tanto en este ejemplo tendríamos $N_s = 31$ y la ecuación (4) se podría reescribir como

$$M = \prod_{i=1}^{4} A_{p_1}^{9} \cdot A_{p_2}^{5} \cdot A_{p_3} \cdot A_{p_4}^{16}, \tag{5}$$

donde A_{p_i} es una matriz $n \times n$ cuyo polinomio característico es el polinomio $p_i(x)$ aplicado al DLFSR.

El polinomio característico $c_M(x)$ de la matriz M determina el período T_M de las secuencias decimadas. Nótese que las N_s secuencias decimadas tienen el mismo polinomio característico [14]. Por tanto, el período T de la secuencia total viene dado por

$$T = T_M \cdot N_s. \tag{6}$$

Por otro lado, el DLFSR establece que las secuencias decimadas se generan con un LFSR de n etapas, luego la complejidad lineal de estas secuencias es n y la complejidad lineal total es

$$LC = n \cdot N_s. \tag{7}$$

IV. Generación de secuencias con mayores períodos y complejidades

A partir de las ecuaciones (6) y (7) se observa que si el polinomio característico $c_M(x)$ fuera primitivo, entonces se alcanzaría el período máximo. En algunos casos, por ejemplo en [12], se puede seleccionar el binomio (N_p, e_i) óptimo para obtener un polinomio $c_M(x)$ primitivo [14]. Sin embargo en otros casos, la única manera de aumentar el período consiste en incrementar el número N_s de secuencias decimadas. Al mismo tiempo, el incremento de N_s también aumenta el valor de la complejidad lineal. Nótese sin embargo que la relación complejidad lineal/período o n/T_M es constante para todo esquema DLFSR; es decir la razón n/T_M es la misma en las secuencias generadas por un DLFSR o por un simple LFSR.

Aunque todos los generadores basados en DLFSR persiguen aumentar el número de secuencias decimadas N_s, se pueden agrupar en distintas categorías en función del modo en que se consigue dicho aumento. Así, tanto en el sistema presentado en [12], como en la configuración del sistema propuesto en [10] cuando $l = 1$, se utiliza un número de polinomios N_p que se aplican siempre en el mismo orden, generando cada

uno de ellos un número de bits constante y determinando un modo de funcionamiento definido por la siguiente expresión

$$M = \prod_{i=1}^{N_p} A_{p_i}^{e_i}, \tag{8}$$

lo que determina que el número total de secuencias decimadas N_s sea

$$N_s = \sum_{i=1}^{N_p} e_i, \tag{9}$$

En otras ocasiones, como en [15], o en el caso general ($l > 1$) de [10], los sistemas DLFSR aumentan N_s utilizando los distintos polinomios siempre en el mismo orden, pero generando cada uno de ellos un número de bits diferente cada vez que se aplican. La expresión que describe su funcionamiento es la siguiente:

$$M = \prod_{i=1}^{L_m} A_{p_{i \bmod N_p}}^{e_i}, \tag{10}$$

siendo L_m la longitud de una secuencia pseudoaleatoria secundaria, lo que determina un número de secuencias decimadas $N_s = \text{mcm}(N_r, N_p)$, donde

$$N_r = \sum_{i=1}^{L_m} e_i, \tag{11}$$

donde $e_i < n$. Por último, propuestas como [1], [4], [9], utilizan un conjunto de polinomios que se aplican siguiendo un orden pseudoaleatorio, pero generando un único bit en cada ocasión. El modelo que define el funcionamiento viene determinado por la siguiente expresión

$$M = \prod_{i=1}^{N_s} A_{p_{z_i}}, \tag{12}$$

donde $1 \leq z_i \leq N_p$, siendo z_i el valor determinado en el instante i por una secuencia pseudoaleatoria de longitud N_s generada por el modulo de control de realimentación del DLFSR. En consecuencia, la matriz M es difícilmente calculable, aunque el período y la complejidad lineal se pueden estimar con las mismas expresiones generales de las categorías anteriores.

De las tres categorías reseñadas, la primera ecuación (8) genera secuencias con una estructura interna que presenta una alta linealidad, lo que facilita su criptoanálisis. La segunda ecuación (10) es la que permite generar períodos mayores. La tercera categoría ecuación (12), aunque presenta períodos menores que las anteriores, tiene una estructura interna mucho más robusta.

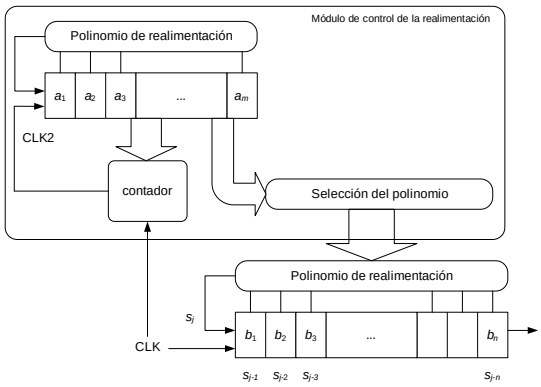

Figura 3. Arquitectura de DLFSR con esquema dinámico variable

IV-A. Generador propuesto basado en DLFSR

En esta sección se propone un diseño de DLFSR genérico, en el que se combinan aspectos de la segunda y tercera categoría, manteniendo unos valores elevados de complejidad lineal y período con una estructura interna más robusta. Utilizando la notación introducida en la sección II, este sistema utiliza un LFSR de n celdas y N_p polinomios, donde cada polinomio $p_i(x)$ genera un número pseudoaleatorio de bits cada vez que se aplica. La regla de selección establece que el orden de aplicación de los polinomios sea pseudoaleatorio, y el módulo de control se compone de un LFSR adicional y un contador, lo que determina un esquema independiente del LFSR principal (Fig. 3). A continuación de describen los componentes del sistema propuesto.

LFSR principal. Es un LFSR de n etapas con N_p polinomios de realimentación primitivos que se irán aplicando siguiendo un orden pseudoaleatorio marcado por el LFSR secundario.

LFSR secundario. Es un LFSR controlado por reloj, de longitud m y polinomio primitivo de grado m, que controla el polinomio de realimentación del LFSR principal mediante a) k_1 de sus bits para la selección de dicho polinomio y b) un contador decreciente que se inicializa con $k_2 \leq \log_2 n$ de sus bits.

Contador. Es un contador que realiza siempre una cuenta atrás completa a partir del valor que determina el LFSR secundario. Al mismo tiempo, el contador controla el reloj de este LFSR secundario. De este modo, cada vez que el contador llega a cero, el LFSR secundario genera un nuevo bit, cambia su estado y reinicializa el contador con un nuevo valor. En ese mismo instante, el polinomio de realimentación del LFSR principal se modifica, puesto que está controlado por k_2 bits del LFSR secundario.

El proceso de generación se detalla tal y como sigue:

- Los LFSRs se inicializan con las correspondientes semillas.
- El contador se inicializa con el estado de k_2 bits del LFSR secundario.
- El LFSR principal comienza a generar bits con el polinomio que determinan k_1 bits del LFSR secundario.

- Simultáneamente el contador comienza la cuenta atrás hasta alcanzar el valor 0. En ese momento, se activa la señal de reloj CLK2 para que el LFSR secundario genere un nuevo bit.
- El nuevo estado del LFSR secundario determina, mediante k_1 bits el nuevo polinomio del LFSR principal, y mediante k_2 el nuevo valor del contador para que inicie la cuenta atrás.

En consecuencia, tanto el orden en el que se aplican los polinomios de realimentación como los bits que genera cada uno vendrán determinados mediante una secuencia pseudo-aleatoria generada por el LFSR secundario. De esta forma, el comportamiento del DLFSR queda determinado por la expresión

$$M = \prod_{i=1}^{L_m} A_{p_{z_i}}^{e_i}, \tag{13}$$

siendo $L_m = 2^m - 1$ la longitud de la secuencia generada por el LFSR secundario, y el número de secuencias decimadas $N_s = N_r$ se calcula aplicando (11). Además, $1 \leq z_i \leq N_p$, siendo z_i el valor determinado en el instante i por k_1 celdas del LFSR secundario.

V. Comparación de resultados

La mejora de la complejidad lineal y del período de las secuencias generadas por el DLFSR propuesto en este trabajo queda patente a través de las expresiones de la sección anterior. Con el fin de ilustrar esta mejora se han realizado diversas comparaciones con algunos de los DLFSR citados previamente. Dado que la relación entre la complejidad lineal y el período de las secuencias generadas por un DLFSR se mantiene constante, las comparaciones se han realizado sobre el número N_s de secuencias decimadas, que es el valor que determina el incremento tanto de la complejidad lineal como del período. Así, el DLFSR(15,6) propuesto por Mita *et al* en [12], perteneciente a la primera categoría de DLFSRs (8), compuesto de un LFSR principal de 16 celdas, otro secundario de 5 celdas y cuatro polinomios de realimentación, presenta un valor $N_s = 2^5 - 1 = 31$ determinado por la longitud de la secuencia generada por el LFSR secundario.

Si configuramos el generador propuesto en este trabajo con los mismos valores, LFSR principal de 16 celdas y secundario de 5 celdas, se tendrían que utilizar $k_1 = 2$ celdas del LFSR secundario para seleccionar los $N_p = 4$ polinomios de realimentación, y $k_2 = 4$ celdas para indicar el número de bits que generará cada polinomio. Es importante recordar que cada polinomio de realimentación no debe generar más de n bits consecutivos, siendo n el grado del polinomio, para evitar posibles criptoanálisis. El número de secuencias decimadas N_s, según indica la ecuación (11) es la suma N_r del valor decimal de todos los estados por los que pasan los k_2 bits. En este caso, $k_2 = 4$ determina la selección de 4 celdas de entre las 5 del LFSR secundario. Por tanto, el período de estas $k_2 = 4$ celdas seguirá siendo $2^5 - 1$, aunque la suma de los estados que se suceden se aproxima por la siguiente expresión

Tabla I
Comparación de resultados

Generador	N_p	N_s
Rakaposhi (192,128)	4	2^{128}
Nuevo DLFSR (192,128)	4	$(2^{128} - 1) \cdot (2^6)$
Mita DLFSR (192,128)	4	$2^{128} - 1$

$$N_s = N_r \leq (2^5 - 1) \cdot (2^{k_2} - 1) = 248, \tag{14}$$

lo que supone una mejora de aproximadamente un orden de magnitud. Sin embargo, estas configuraciones no se corresponden con los valores de las implementaciones reales. Por ello, se ha realizado una comparación (tabla I) con el DLFSR utilizado en el cifrador Rakaposhi [4] que utiliza un LFSR principal de 192 celdas y un NLFSR secundario de 128. Al pertenecer este DLFSR a la segunda categoría (10), el número de secuencias decimadas N_s se corresponde directamente con la longitud de la secuencia generada por el NLFSR, es decir, $N_s = 2^{128}$. Si se configura el DLFSR propuesto con un LFSR principal de 192 celdas y un LFSR secundario de 128, se necesitarían $k_1 = 2$ celdas para seleccionar los $N_p = 4$ polinomios y $k_2 = 7$ celdas para indicar el número de bits consecutivos que cada polinomio debe generar. En consecuencia, como indica la tabla I, el número de secuencias N_s se puede aproximar como

$$N_s = N_r \leq (2^{128} - 1) \cdot (2^{k_2 - 1}) \tag{15}$$

El DLSFR propuesto en este trabajo incrementa N_s, y por tanto la complejidad lineal y el período, en un factor de $(2^{k_2} - 1)$ con respecto a la que se obtiene con el cifrador Rakaposhi. Por último, para completar la comparación, se analiza el DLFSR genérico propuesto en [12] para los mismos valores del cifrador Rakaposhi. Se obtiene $N_s = 2^{128} - 1$, que es muy similar al valor obtenido para el cifrador Rakaposhi. La diferencia reside en la estructura interna de las secuencias generadas en [12], que permite a un atacante criptoanalizarlas con facilidad.

VI. Conclusiones

En la actualidad numerosos generadores de secuencia cifrante para uso criptográfico pertenecen al grupo de generadores DLFSR.

En este trabajo se ha desarrollado un nuevo tipo de generador DLFSR que mejora la complejidad lineal y el período de las secuencias producidas. Para ello, se ha partido de una clasificación de los sistemas basados en DLFSR en función de la técnica empleada para aumentar N_s (el número de secuencias decimadas que conforman la secuencia entrelazada global); se ha modelado su funcionamiento a partir del modelo propuesto en [14]; y se ha propuesto un nuevo tipo que combina las ventajas de los anteriores, mejorando complejidad lineal y período y disminuyendo la linealidad de la estructura interna del generador.

El esquema propuesto está formado por dos LFSRs y un contador combinados de manera que el reloj del LFSR principal está controlado por el contador, que a su vez está controlado por el LFSR secundario. El efecto que se consigue es que los polinomios de realimentación se apliquen siguiendo un orden pseudoaleatorio y que cada uno de estos polinomios genere un número de bits consecutivos determinados también de forma pseudoaleatoria.

Por último se han comparado los valores del parámetro N_s que utilizan algunos de los generadores basados en DLFSR que se han propuesto recientemente, como es el caso del cifrador Rakaposhi.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MICINN en el marco del proyecto "TUERI: Tecnologías segUras y Eficientes para las Redes inalámbricas en la Internet de las cosas con aplicaciones en transporte y logística", TIN2011-25452; y por la Universidad de Málaga, Andalucía Tech.

REFERENCIAS

[1] S. Babbage, M. Dodd, "The MICKEY Stream Ciphers," in *New Stream Cipher Designs. The eSTREAM Finalists*, M. Robshaw, O. Billet, Eds. LNCS 4986, Springer-Verlag, 2008, pp. 191–209.

[2] A. Bogdanov, B. Preneel, V. Rijmen, "Security Evaluation of the K2 Stream Cipher," Internal report, Katholieke Universiteit Leuven, ESAT/SCD-COSIC, March 2011.

[3] W. Che, H. Deng, X. Tan, J. Wang, "A Random Number Generator for Application in RFID Tags," in *Networked RFID Systems and Lightweight Cryptography*. Springer: Berlin/Heidelberg, Germany, 2008, Chapter 16, pp. 279–287.

[4] C. Cid, S. Kiyomoto, J. Kurihara, "The RAKAPOSHI Stream Cipher," in *Information and Communications Security*, LNCS 5927[C], Springer-Verlag, 2009, pp. 32–46.

[5] L. Ding, J. Guan, "Cryptanalysis of Mickey family of stream ciphers," *Security and Communication Networks*, vol. 6 (8), pp. 936–941, 2013.

[6] S.W. Golomb, "Shift-Register Sequences," revised edition, Aegean Park Press, Laguna Hill, California, 1982.

[7] G. Gong, "Theory and Applications of q-ary interleaved sequences," *IEEE Transactions on Information Theory*, vol. 41 (2), pp. 400–411, 1995.

[8] S. Hellebrand, J. Rajskia, S. Tarnick, S. Venkataraman, B. Courtois, "Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers," *IEEE Trans. Comput.*, vol. 44, pp. 223–233, 1995.

[9] S. Kiyomoto, T. Tanaka, K. Sakurai, "K2: A stream cipher algorithm using dynamic feedback control," in *Proceedings of SECRYPT*, J. Hernando, E. Fernández-Medina, M. Malek, Eds. INSTICC Press, 2007, pp. 204–213.

[10] J. Meliá-Seguí, J. García-Alfaro, J. Herrera-Joancomartí, "J3Gen: A PRNG for Low-Cost Passive RFID," *Sensors*, vol. 13, pp. 3816– 3830, 2013.

[11] J. Meliá-Seguí, J. García-Alfaro, J. Herrera-Joancomartí, "A practical implementation attack on weak pseudorandom number generator designs for EPC Gen2 Tags," *Wirel. Pers. Commun.*, vol. 59, pp. 27– 42, 2011.

[12] R. Mita, G. Palumbo, S. Pennisi, M. Poli, "Pseudorandom bit generator based on dynamic linear feedback topology," *Electronic Letters*, vol. 38 (19), pp. 1097– 1098, 2002.

[13] M.A. Orumiehchiha, J. Pieprzyk, E. Shakour, R. Steinfeld, "Security Evaluation of Rakaposhi Stream Cipher," in *Information Security Practice and Experience*, 9th International Conference, ISPEC 2013, R. Deng, T. Feng, Eds. LNCS 7863, Springer-Verlag, 2013, pp. 361–371.

[14] A. Peinado, A. Fúster-Sabater, "Generation of pseudorandom binary sequences by means of LFSRs with dynamic feedback," *Mathematical and Computer Modelling*, vol. 57 (11-12), pp. 2596–2604, 2013.

[15] A. Peinado, J. Munilla, A. Fúster-Sabater, "Improving the Period and Linear Span of the Sequences Generated by DLFSRs," *7th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2014*, Bilbao, Spain, 25th-27th June, 2014

[16] P. Rosinger, B. Al-Hashimi, N. Nicolici, "Dual multiple-polynomial LFSR for low-power mixed-mode BIST," *IEEE Proc. Comput. Digital Tech.*, vol. 150 , pp. 209–217, 2003.