Implementación hardware de un cifrador basado en una aplicación caótica

Diego Torre, Miguel García, Carlos Sánchez-Azqueta, Concepción Aldea, Santiago Celma

Grupo de Diseño Electrónico (GDE)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: 698323@unizar.es

Resumen

Este trabajo se centra en la síntesis y verificación posterior de un cifrador de flujo basado en la aplicación caótica *Skew Tent Map*. El trabajo concluye con el estudio de posibles modificaciones con el fin de mejorar la calidad de las secuencias aleatorias generadas.

Introducción

En los últimos años, las redes de comunicaciones y la transferencia de datos se han desarrollado de forma exponencial, surgiendo la necesidad del estudio de sistemas criptográficos para asegurar la confidencialidad e integridad en el intercambio de información.

La seguridad de los sistemas criptográficos está fuertemente relacionada con la aleatoriedad, ya que la salida de estos sistemas debe ser vista por un adversario como una secuencia de valores aleatorios, sin aportar claves sobre la preciada información [1]. Una tendencia creciente es utilizar números aleatorios para encriptar el mensaje, por lo tanto, la calidad de la protección dependerá directamente de la calidad de los números aleatorios empleados.

Este trabajo ha apostado por utilizar un PRNG (Pseudorandom Number Generator) basado en una aplicación caótica concreta, el *Skew Tent Map*, debido a que la órbita del sistema caótico tiene la propiedad de ser irregular, aperiódica e impredecible, y junto a la sensible dependencia con las condiciones iniciales resulta muy adecuado para la generación de números pseudoaleatorios [2]. Finalmente, se ha realizado un análisis de la calidad de las secuencias aleatorias generadas a partir de diferentes estrategias.

Este trabajo ha sido parcialmente apoyado por MINECO-FEDER (TEC2014-52840-R y TEC2017-85867-R) y una beca FPU a M. Garcia (FPU/03523).

Skew Tent Map

La aplicación clásica *Tent Map* se define a partir de la siguiente ecuación:

$$x_{i+1} = \begin{cases} \gamma x_i, & x_i \in [0, 0.5] \\ \gamma(1 - x_i), & x_i \in (0.5, 1] \end{cases} \quad (1)$$

donde $\gamma \in [0,2]$ es un parámetro caótico que, junto al valor inicial de la secuencia, $x_0 \in [0,1]$, conforma la clave secreta que tanto el transmisor como el receptor necesitan conocer para encriptar y desencriptar el mensaje respectivamente.

El *Tent Map* exhibe comportamiento caótico para ciertos valores de γ, pero no todos. Debido a ello, se ha utilizado una variación de ese mapa caótico, el *Skew Tent Map*, cuyo comportamiento es caótico para todos los posibles valores de los parámetros iniciales [3]. Sus ecuaciones son:

$$x_{i+1} = \begin{cases} \frac{x_i}{\gamma}, & 0 \le x_i \le \gamma \\ \frac{(1-x_i)}{1-\gamma}, & \gamma < x_i \le 1 \end{cases} \quad (2)$$

siendo tanto γ como $x_n \in [0,1]$. Además, el *Skew Tent Map* produce una salida uniformemente distribuida, que es una importante propiedad para la generación de números aleatorios. Esta aplicación ha sido adaptada a aplicaciones de encriptado y digitalizada en forma de circuito integrado mediante el lenguaje de descripción hardware *Verilog*.

Implementación

El criptosistema desarrollado pertenece a los cifradores simétricos, ya que emplea la misma clave para encriptar y desencriptar. El circuito completo ha sido diseñado usando la tecnología digital nanométrica CMOS TSMC180 en el entorno de simulación *Cadence*, donde se han desarrollado e implementado sus diferentes bloques constitutivos. Dado que la arquitectura de la aritmética digital que

se ha utilizado es la de punto fijo, la precisión del sistema viene dado por la cantidad fija de dígitos después del punto decimal. Se ha empleado una precisión de 32 bits para cada variable, por lo que, como máximo (idealmente), la periodicidad del sistema puede ser 2^{32}.

La configuración software del circuito consta de cuatro módulos: el primero de ellos se encarga de realizar las operaciones matemáticas de la aplicación caótica; luego existe un módulo de control que asegura el correcto funcionamiento y evita solapamientos; el tercero extrae el bit menos significativo (LSB) de la iteración de la aplicación; y, por último, uno superior que interconexiona al resto. Utilizamos el LSB de la iteración para encriptar ya que se entiende que éste es el que con más facilidad sufre cambios (y por tanto más aleatorio), y además limitamos la información accesible sobre el sistema.

Resultados

En la figura 1 se puede ver la vista del layout del circuito integrado. Cabe destacar que el elevado número de pads se debe a que se ha diseñado de forma que las condiciones iniciales de la aplicación (γ y x_0), compuestas por 32 bits cada una, se introduzcan externamente y haya posibilidad de cambiar las secuencias emitidas.

Con el objetivo de analizar la seguridad de este algoritmo, las secuencias generadas han sido sometidas a la batería de test de aleatoriedad del Instituto Nacional de Estándares y Tecnología (NIST) SP 800-22. En la figura 2a, se aprecia que el cifrador implementado basado en el *Skew Tent Map* no supera las pruebas. Este hecho se puede deber a la existencia de órbitas caóticas de corta longitud. Por este motivo se han explorado dos estrategias diferentes con el fin de garantizar la calidad de las secuencias.

El primero de ellos consiste en perturbar las órbitas con un *Linear Feedback Shift Register* (LFSR), de forma que se realiza la operación XOR entre el LSB de cada x_i con un bit generado por el LFSR. La segunda modificación consiste en la generación de nuevas llaves cada 1000 bits, de acuerdo con la aplicación caótica *Tent Map*. La figura 2b y 2c muestra que ambas técnicas superan el valor de significancia de los test, indicando que generan secuencias propias de un generador aleatorio.

Conclusiones

Este trabajo ha llevado a cabo la implementación sobre silicio de un cifrador de flujo. Se han estudiado dos formas de garantizar la calidad de las secuencias, superando ambas satisfactoriamente los test de aleatoriedad del NIST.

REFERENCIAS

[1]. GARCIA-BOSQUE, M., PÉREZ, A., SÁNCHEZ-AZQUETA, C., and CELMA, S. A new simple technique for improving the random properties of chaos-based cryptosystems. *AIP Advances*, vol. 8, no. 3, pp. 1-11, 2018.

[2]. GARCIA-BOSQUE, M., PÉREZ, A., SÁNCHEZ-AZQUETA, C., and CELMA, S. Application of a MEMS-Based TRNG in a Chaotic Stream Cipher. *MDPI Sensors*, vol 7, no. 3, pp 1-15, 2017.

[3]. GARCIA-BOSQUE, M., SÁNCHEZ-AZQUETA, C., and CELMA, S. Lightweight Ciphers Based on Chaotic Map-LFSR Architectures. *12th IEEE Conference on PhD Research in Microelectronics and Electronics (PRIME 2016)*. June 2016.

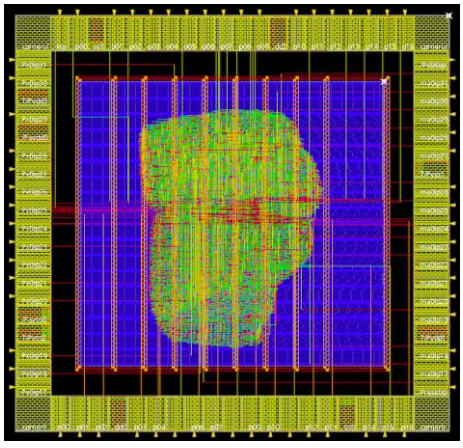

Figura 1. Layout del cifrador implementado

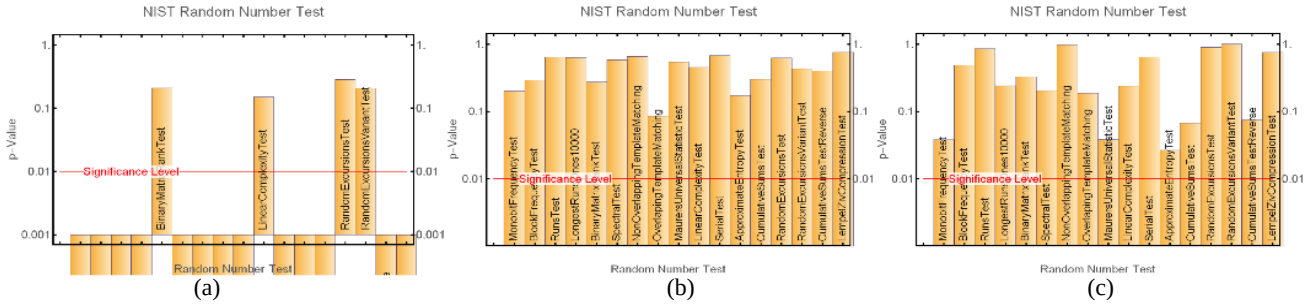

Figura 2. Resultados del NIST: (a) solo *Skew Tent Map*, (b) incluyendo LFSR y (c) incluyendo *tent map* generador de llaves.