Resumen: Cycle detection for secure chaos-based encryption.

Marcos Daniel Calderón Calderón
Maestría en Ciencias de la Computación
Centro de Investigación en Matemáticas (CIMAT)
Guanajuato , Gto.
marcos.calderon@cimat.mx

***Resumen*—La generación digital de caos presenta varias limitaciones que afectan la vulnerabilidad de sistemas de encriptación por caos. Perturbaciones periodicas de los parametros caóticos y las variables de estado han sido utilizadas para tratar con estas limitaciones.**

I. Introducción.

Sistemas Digitales Caóticos (DCS) han sido propuestos como una alternativa viable a la encriptación clásica (AES, DES) para la comunicación segura en texto, imágenes y más recientemente en aplicaciones de voz y de video. Los DCS se representan por una máquina de estado finito, lo cual reduce sus propiedades caóticas (ciclos caóticos, función de distribución de probabilidad, ergodicidad, etc.) independientemente de sus trayectorias y condiciones iniciales. Lo que hay que preservar en los DCS es la sensibilidad a la s condiciones iniciales, trayectorias impredecibles y propiedades estadísticas tales como distribución de probabilildad invariante y exponente de Lyapunov.

Una solución común para para mejorar el comportamiento de los DCS es perturbar de manera periódica la variable de estado para generar una trayectoria totalmente diferente con una longitud de ciclo caótica más grande (L). Para que esto ocurra, la periodicidad de la perturbación (P) debe ser aplicada de manera apropiada. Si la perturbación es aplicada tarde $(P >> L)$, la seguridad del criptosistema puede estar en riesgo al reutilizar las viejas trayectorias, si la perturbación es aplicada muy pronto $(P << L)$, que es lo más común, puede ocurrir alguna de las siguientes opciones:

- L se incrementa de manera proporcional a $\sigma * \Delta * L$ (donde σ es un entero positivo y Δ es un entero no tan pequeño que L), pero puede no satisfacer la necesidad actual del criptosistema (esto ocurre particularmente en sesiones multimedios de términos grandes tal como voz o video).
- Otra cosa que puede ocurrir es que el tamaño del ciclo puede terminar teniendo el mismo periodo que la perturbación (L=P), como se menciona en control de teoria del caos.Para obtener el mayor valor para L para un mapa caótico particular, la periodicidad del paerturbación y la longitud del ciclo caótico debe ser aproximadamente la misma $(P \sim L)$. Como L no puede ser analíticamente conocida apriori, la periodicidaqd de la perturbación en esquemas actuales es aplicada sin cuestionarla. Nosotros proponemos una cuantificación del tamaño del ciclo del sistema caótico para reducir el riesgo de perturbaciones periódicas y mejorar de manera considerable las propiedades del sistema caótico bajo ciertas consideraciones.

Hay muchos métodos numéricos en la literaruta dirigidos a encontrar un conjunto completo de Órbitas Periódicas Inestables , pero éstas son imprácticas para criptosistemas en tiempo real donde la velocidad del procesamiento y detecciones de ciclo grandes son cruciales.

II. Unrestricted Search Algorithm.

La idea general de este algoritmo es localizar la trayectoria caótica para un rango amplio de sistemas dinámicos usando un muestreo inicial de un intervalo P_0. Si no se encontraron ciclos durante las primeras iteraciones, este gradualmente cambia sus espectativas de encontrar ciclos largos al incrementar el intervalo de muestreo y elegir un espacio el doble de lo elegido al principio, este proceso se repite hasta que un ciclo es encontrado. Una de las principales ventajas del método USA es que sólo son considerados R puntos en el ciclo de búsqueda, independientemente del tamaño de ciclo. El algoritmo se divide en pasos, y cada paso es dividido en rondas con diferentes intervalos de muestreo. El procedimiento se puede resumir de la siguiente manera:

1. *Proceso de inicialización.* El primer paso toma el conjunto de trayectorias $\left\{X_{l=n/p_{k=0}}, l \in (0, 1, ...P_0R)\right\}$ (un punto a la vez) y las acomoda en una tabla hash en cada P_0 iteraciones hasta que un total de R puntos son acomodados. Los puntos intermedios $X_{l=n/p_0}, l \notin (0, 1, ...P_0R)$ son únicamente verificados si existen o no en la tabla hash, pero no son almacenados; si un punto intermedio o un punto de la mestra ya existe en la tabla, un ciclo ha sido encontrado. Y en este caso asignamos $k \leftarrow 1$.
2. *Step k:* En la primer ronda de el paso k-ésimo, los $\frac{R}{2}$ puntos de la muestra con índices impares son reemplazados (siguiendo estrictamente un orden de crecimiento de I) por los siguientes N puntos que se muestrean en cada P_k iteraciones. La nueva señal muestreada R tiene ahora dos diferentes intervalos de

muestreo, P_k en la primer mitad y P_{k-1} en la segunda mitad donde $P_k = 2P_{k-1}$=$2^k P_0$ (el tamaño de R es constante, sólo cambian los datos). Este procedimiento se repite para la segunda parde del array R, teniendo $\frac{3}{4R}$ con un periodo P_K y $\frac{1}{4R}$ con periodo P_{k-1} (segunda ronda).El número total de rondas que se necesitan para tener un intervalo uniforme de P_k es $[\log_2 R + 1]$, lo cual marca el final del primer paso. Y como en el pasoprevio, cada punto muestreado o intermedio es verificado si está o no en la tabla hash.

3. Si un ciclo aún no ha sido encontrado, se hace $k \leftarrow k + 1$, $P_k = 2^k P_0$ y regresmos al paso 2.