# Generadores de números pseudo aleatorios acoplados y sus aplicaciones en criptografía

**José Manuel Albornoz[1,2*], Antonio Parravano[2]**

[1]Departamento de Electrónica y Comunicaciones, Escuela de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela. [2]Postgrado de Física Fundamental, Departamento de Física, Facultad de Ciencias, Universidad de Los Andes, Mérida, Venezuela.
* albornoz@ula.ve

**Resumen**

En este trabajo se estudia el comportamiento de un sistema de generadores de números pseudo aleatorios (generadores de congruencia lineal, GCLs) en condiciones de acoplamiento mutuo. Se muestra cómo el acople mutuo mejora las propiedades de estos generadores, haciéndolas más apropiadas para aplicaciones criptográficas. Un algoritmo de cifrado que involucra un sistema de GCLs acoplados es propuesto.

**Palabras clave**: criptografía, generadores de números pseudo aleatorios, generadores de congruencia lineal.

# Coupled pseudo random number generators and their cryptographic applications

**Abstract**

The behavior of a system of pseudo random number generators (linear congruential generator, LCGs) is studied under mutual coupling conditions. The results of the study show that this coupling improves the properties of such generators, which in isolation are not suited for cryptographic applications. A ciphering algorithm using a system of LCGs is proposed.

**Key words**: cryptography, pseudo random number generators, linear congruential generators.

## 1. Introducción

Los generadores de números pseudo aleatorios son de vital importancia en muchas aplicaciones criptográficas para la generación de claves y códigos de acceso; sin embargo, en este trabajo los utilizamos como parte de un sistema de cifrado. Uno de los generadores más antiguos y sencillos es el generador de congruencia lineal (GCL) [1], en el que secuencias de números pseudo aleatorios son producidos a partir de la siguiente relación de recurrencia

$$x_{n+1} = (Ax_n + B) \bmod M \,, \qquad [1]$$

En esta expresión *A, B* y *M* tienen ciertos valores fijos, en tanto que $x_0$ es un número inicial o semilla. En la implementación computacional de este tipo de generador las constantes *A* y *B* son escogidas para que se produzca desborde en la mayoría de las iteraciones; este desborde es equivalente a la operación *mod(x,M)*[1], tal como ésta es definida en lenguajes de programación como C o FORTRAN. Estos generadores se denotan como (*M*, *A*, *B*, $x_0$); en el caso particular en el que $B = 0$ se tiene un generador de congruencia lineal multiplicativo.

Este tipo de generador es computacionalmente rápido y de fácil implementación; sin embargo es bien conocido que sus propiedades no son ideales: un GCL produce una secuencia de valores que se repiten con un período que a lo sumo es *M-1*, y si los valores de *A* y *B* no son los apropiados, con frecuencia dicho período es mucho menor: por ejemplo, con GCL(10, 7, 7, 7) se obtiene la secuencia 7, 6, 9, 0, 7, 6, 9, 0, ....[2]. Por otra parte, las secuencias producidas por un GCL son muy sensibles a cambios en sus parámetros, lo cual es una propiedad útil, sin embargo un multiplicador *A* inadecuado producirá una secuencia de valores altamente correlacionados.

Otro problema de los GCL es que si se escoge como valor de *M* una potencia de dos (lo cual ocurre precisamente cuando se emplea el desborde en una palabra de *N* bits), los bits menos significativos de cada uno de los números de la secuencia generada se repetirán con un período mucho menor que el período de la secuencia.

Una propiedad de los GCLs que los hace particularmente inadecuados para aplicaciones criptográficas es su predictibilidad. Por ejemplo, el mapa de retorno *n*-dimensional de un GCL (*M*, *A*, *B*, $x_0$) muestra que los puntos generados yacerán a lo sumo en $(n!M)^{1/n}$ hiperplanos, un fenómeno conocido como el "efecto Marsaglia" [3]. Un ejemplo de este tipo de efecto es el asociado al generador RANDU ($2^{31}$, 65539, 0, $x_0$), distribuido por IBM en la década de 1960. La Figura 1 muestra la disposición espacial de tripletas generadas a partir de la secuencia producida por este generador: teóricamente, los puntos asociados a estas tripletas yacerán a lo sumo en 2344 hiperplanos; sin embargo, para el valor particular de *A* empleado en RANDU se observa que las tripletas yacen en 15 hiperplanos, una cantidad mucho menor que el máximo teórico.

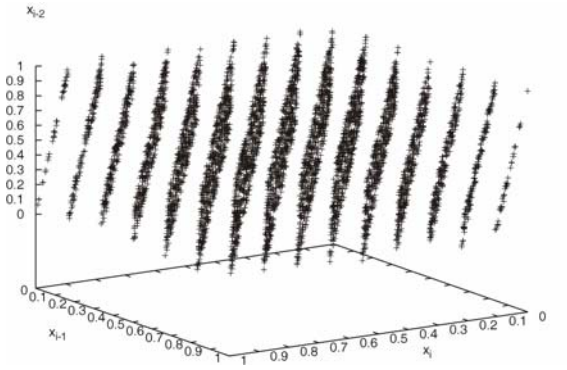


Figura 1. Mapa de retorno tridimensional para RANDU ($2^{31}$, 65539, 0, $x_0$).

La presencia de esta estructura en las secuencias producidas por un GCL hacen

---

[1] De manera aproximada, en la implementación digital de un GCL se tiene

$$x_{n+1} = (Ax_n + B) \bmod 2^{N-1}, \qquad [2]$$

donde *N* es la longitud de la palabra binaria empleada; en consecuencia $M = 2^{N-1}$. Nótese que en la representación binaria de un entero el bit más significativo denota el signo y los restantes *N-1* bits denotan el valor absoluto de un número comprendido entre 0 y $2^{N-1}$; en consecuencia, la operación que tiene lugar con el desborde de una palabra binaria de N bits no es en rigor *mod* $2^{N-1}$, ya que los valores producidos estarán comprendidos entre $-2^{N-1}$ y $2^{N-1}-1$

factible la determinación de ($M$, $A$, $B$, $x_0$) a partir del análisis de una secuencia parcial de los valores producidos por el generador. Es esta última característica lo que hace que un GCL sea inadecuado para muchas aplicaciones criptográficas. Ejemplos de ataques sobre secuencias parciales producidas por un GCL pueden encontrarse en [4] y [5]. Sin embargo, en el presente trabajo se muestra que puede diseñarse un sistema criptográfico seguro basado en GCLs efectuando dos modificaciones. La primera es colapsar el espacio de fase de los GCLs desde $n = (-2^{N-1},...,0,...2^{N-1}-1)$ hasta $n = (0,1)$. La segunda modificación consiste en acoplar varios GCLs para, por una parte, disminuir las correlaciones en las secuencias de cada generador, y por otra parte, aumentar el tamaño de los segmentos de texto plano que pueden ser cifrados en cada iteración del sistema.

## 2. Colapso del Espacio de Fases

Para $x_0$, $M$, $A$ y $B$ enteros el GCL ($M$, $A$, $B$, $x_0$) produce una secuencia de números enteros $x_n$ que está contenida en el conjunto de los números naturales $\{0,.., M-1\}$. Por medio de la aplicación de un umbral $x_U$ la secuencia $x_{1\text{-}end} \equiv (x_1, .., x_{end})$ puede ser convertida en la secuencia binaria $b_{1\text{-}end} \equiv (b_1, .., b_{end})$. Esto es

$$b_n = 0 \quad \textbf{si} \quad x_n < x_U$$
$$b_n = 1 \quad \textbf{si} \quad x_n > x_U, \qquad [3]$$

así, la dimensión del espacio de fase de la secuencia es reducida de $\sim M$ a 2, lo que representa una reducción drástica ya que $M$ es usualmente del orden de $2^{31}$. Ahora, ¿cuán más difícil es determinar los parámetros $x_0$, $M$, $A$ y $B$ a partir de la secuencia $b_{1\text{-}end}$ que de la secuencia $x_{1\text{-}end}$? En otras palabras, ¿cuál es la longitud mínima necesaria de la secuencia binaria $d_{min}$ para determinar unívocamente los parámetros del GCL que luego del "colapso" genera la secuencia? Una estimación del tamaño mínimo de $d_{min}$ puede obtenerse de la siguiente manera: Sea $\{C_i\}$ el subconjunto de semillas que generan el bit $b_i$ después de la $i$-ésima iteración del GCL, y $\{S_i\}$ el subconjunto de semillas que generan la secuencia $b_{1\text{-}i} = (b_1,....., b_i)$. Así,

$$\{S_i\} = \{C_1\} \cap \{C_2\} \cap ..... \cap \{C_i\}. \qquad [4]$$

Para el caso en que $x_u = M/2$ (en el caso del GCL descrito por (2), $x_u = 0$), cada uno de los subconjuntos $C_i$ contendrá aproximadamente la mitad de todas las semillas contenidas en el subconjunto $C_{i-1}$. Asumiendo que los valores generados por el GCL son equiprobables, que el período del GCL es el máximo ($\sim M$) y que los bits generados al aplicar (2) son independientes entre sí, la fracción $|S_i|/M$ de semillas en el subconjunto $\{S_i\}$ estará dada aproximadamente por $(1/2)^i$. Así, una estimación de la longitud mínima $d_{min}$ se obtiene con la condición $|S_{dmin}| = 1$, por lo que $d_{min} = log_2 M = N-1$. En consecuencia, sería necesario disponer de una secuencia de al menos $N-1$ bits en el caso del generador descrito por (2) y (3) para poder determinar unívocamente la semilla que produce una secuencia prescrita de bits. Cabe mencionar que dado que el GCL es un sistema dinámico pseudo aleatorio, hay secuencias $(b_1,...., b_i)$ que no ocurrirán (i.e. $|S_i| = 0$).

Asumiendo una representación binaria de 32 bits de $A$, $B$, $M$ y $X_0$, la búsqueda exhaustiva de estos parámetros supondría ensayar $2^{128} \approx 10^{39}$ combinaciones. Suponiendo $A$, $B$ y $M$ conocidos, en principio se necesitarían $\sim 2^{32}$ pruebas para encontrar la semilla que da origen a una determinada secuencia de bits. Aproximadamente la mitad de las semillas

serán descartadas en la primera iteración, la mitad de la mitad restante será descartada en la segunda iteración, y así sucesivamente. El número promedio de iteraciones $I_{prom}$ del mapa descrito por (2) que sería necesario realizar para encontrar la semilla que produce una determinada secuencia de bits será entonces

$$I_{prom} = 2^{32}\sum_{i=0}^{d_{\min}}\left(\frac{1}{2}\right)^{i} = 2^{32}\left[2-\left(\frac{1}{2}\right)^{d_{\min}}\right] \quad [5]$$

Procedimientos no exhaustivos para determinar los parámetros $A$, $B$, $M$ y $X_0$ del GCL que produce la secuencia $b_{1\text{-}dmin}$ podrían ser desarrollados para reducir el espacio de búsqueda. El desarrollo de estos métodos no exhaustivos está fuera de los objetivos del presente trabajo. Sin embargo, como veremos más adelante, el uso de GCLs acoplados, por una parte, dificulta el desarrollo de tales métodos y por otra, aumenta el número de parámetros a ser determinados.

## 3. GCLs Acoplados

Una técnica sencilla que permite utilizar GCLs en aplicaciones criptográficas es la de emplear un conjunto de $N$ GCLs acoplados:

$$x_{n+1}^{k} = \left(1-\left\langle \varepsilon_{kj}\right\rangle\right) f\left(x_n^k\right) + \sum_{j\neq k}^{N}\varepsilon_{kj} f\left(x_n^j\right), \quad [6]$$

donde

$$\left\langle \varepsilon_{kj}\right\rangle = \frac{1}{N-1}\sum_{j\neq k}^{N}\varepsilon_{kj} \quad [7]$$

En esta expresión $f(\bullet)$ representa la operación descrita por (1) para los parámetros $A^k$, $B^k$, $C^k$, $M^k$, y $\varepsilon_{kj}$ un parámetro de acople entre los GCLs. La idea de utilizar generadores acoplados ha sido anteriormente explorada en el contexto de la utilización de un conjunto de mapas caóticos acoplados como mecanismo de cifrado [6].

La Figura 2 muestra la disposición espacial de las tripletas generadas por una de las salidas de 2 GCLs RANDU idénticos acoplados con $\varepsilon_{12} = \varepsilon_{21} = 0.5$: la estructura previamente hallada para el caso de un solo GCL (Fig. 1) ya no está presente. Esto sugiere una mejora en las propiedades de los GCL acoplados, ya sea a través de una distribución de los valores generados sobre un mayor número de hiperplanos, o a través de la eliminación/reducción de la correlación entre los valores generados.

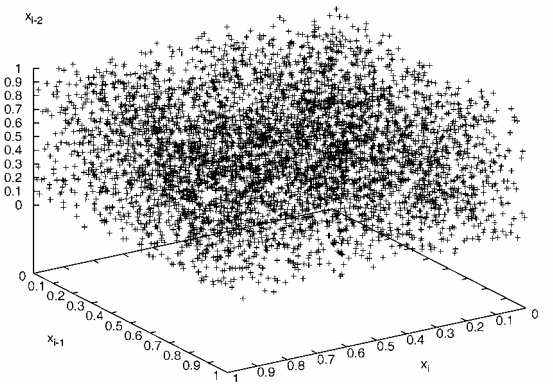


Figura 2: Mapa de retorno tridimensional para el caso de 2 GCLs RANDU ($2^{31}$, 65539, 0, $x_0$) idénticos acoplados.

Es un hecho conocido que la realización computacional de un mapa caótico presenta periodicidad debido a la precisión finita ofrecida por el computador, y que el período asociado a tal realización aumenta de acuerdo a una ley de potencias de la forma $10^{\alpha N}$ para un conjunto de $N$ mapas caóticos acoplados [7]. En este sentido, un GCL puede considerarse como la versión discreta del mapa de Bernoulli [8], y las ecuaciones (6) y (8) como la versión discreta de la expresión que describe a un conjunto de mapas acoplados, por lo que cabría esperar que el acople también incremente el período de un GCL, afectando su mapa de retorno.

A fines de examinar la influencia del acople sobre los valores generados examinaremos el caso más simple posible, en el cual dos GCLs idénticos son acoplados simétricamente (i.e. $A^1 = A^2 = A$,..., y $\varepsilon_{12} = \varepsilon_{21} = \varepsilon$). En este caso, la ecuación (6) toma la forma

$$\begin{cases} x_{n+1}^1 = (1-\varepsilon) f(x_n^1) + \varepsilon f(x_n^2) \\ x_{n+1}^2 = \varepsilon f(x_n^1) + (1-\varepsilon) f(x_n^2), \end{cases} \quad [8]$$

con

$$\begin{cases} b_n^k = 0 \quad \textbf{si} \quad x_n^k < x_U \\ b_n^{ki} = 1 \quad \textbf{si} \quad x_n^k > x_U \end{cases}, \text{ con k = 1,2} \quad [9]$$

La secuencia de pares de bits ($b_n^1$, $b_n^2$) producida dependerá de los parámetros $A$, $B$ y $M$ del GCL empleado, de la magnitud del acople $\varepsilon$, y de los valores de las semillas $x_0^1$ y $x_0^2$ usadas para cada generador. A fines de determinar cómo la correlación entre $b_n^1$ y $b_n^2$ es afectada por el número de iteraciones y el valor del acople $\varepsilon$, se realizaron pruebas en las que se exploró el espacio de valores de las semillas ($x_0^1$, $x_0^2$) en los intervalos ($x_{00}^1$ , $x_{00}^1$ + 200), ($x_{00}^2$, $x_{00}^2$ + 200) para un valor fijo de $\varepsilon$, examinándose los valores ($b_n^1$, $b_n^2$) tomados por el par de secuencias después de un cierto número de iteraciones. La Fig. 3 muestra el conjunto de pares de semillas ($x_0^1$, $x_0^2$) que produjeron la secuencia '11' después de 25 iteraciones con $\varepsilon = 0$ y $x_0^1 \neq x_0^2$. La regularidad del patrón observado es una indicación de la correlación entre las secuencias generadas por ambos generadores. En contraste, la Fig. 4 muestra el resultado para $\varepsilon = 10^{-5}$ después de 25 iteraciones: la introducción del acople implica la reducción de la correlación entre las secuencias de bits $b_n^1$ y $b_n^2$; esta reducción aumenta a medida que progresa el número de iteraciones. La Fig. 5 muestra el conjunto de semillas que produjeron el par '11' para un acople asimétrico $\varepsilon_{12} = 0.4$, $\varepsilon_{21} = 0.15$; esta vez se apreció la desaparición de toda regularidad en tan sólo 10 iteraciones.

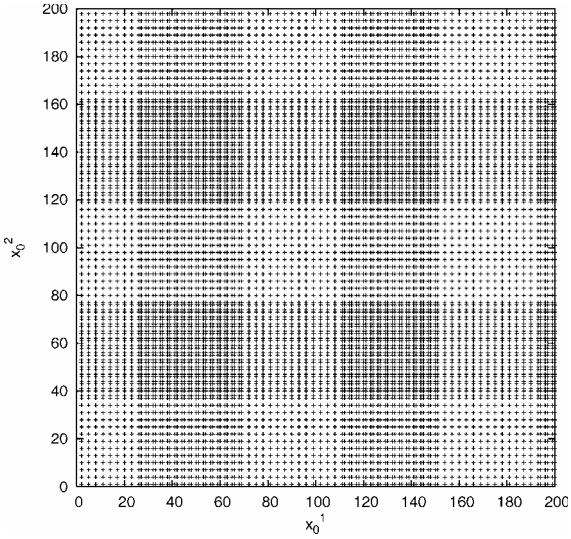


Figura 3: Conjunto de semillas que producen en 25 iteraciones el par ($b_{25}^1$, $b_{25}^2$) = (1,1) para $\varepsilon = 0$.

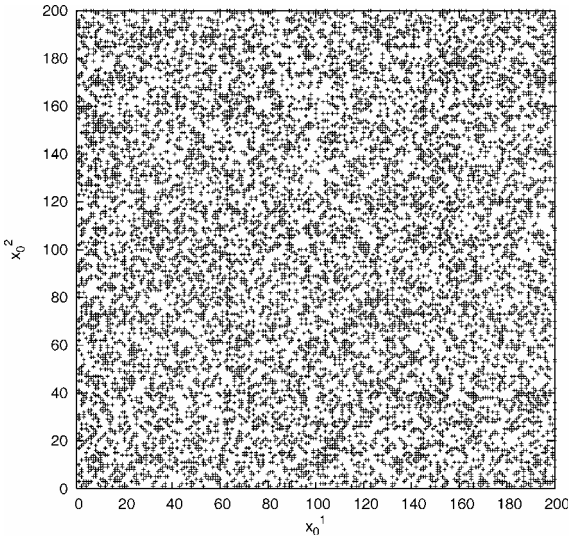


Figura 4: Conjunto de semillas que producen en 25 iteraciones el par ($b_{25}^1$, $b_{25}^2$) = (1,1) para $\varepsilon = 10^{-5}$.

.

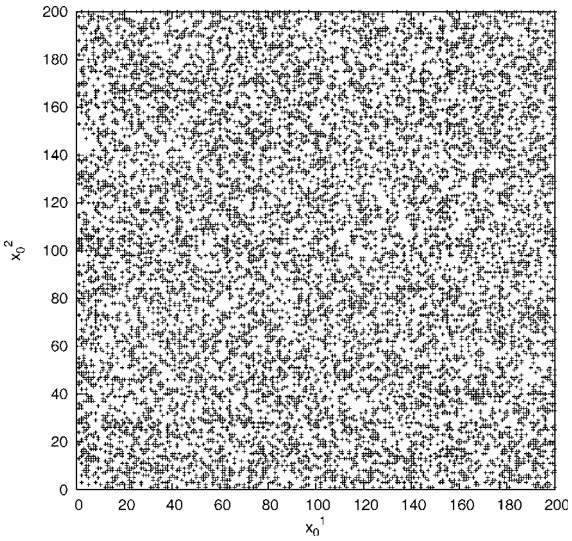


Figura 5: Conjunto de semillas que producen en 10 iteraciones el par ($b_{10}^1$, $b_{10}^2$) = (1,1) para $\varepsilon_{12} = 0.3$, $\varepsilon_{21} = 0.15$.

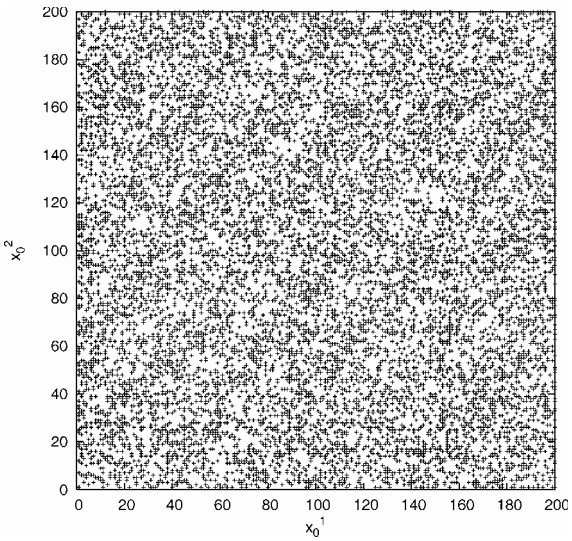


Figura 6: Conjunto de semillas que producen en 4 iteraciones el par ($b_4^1$, $b_4^2$) = (1,1) para $\varepsilon_{11} = 0.23745$, $\varepsilon_{12} = 0.77362$, $\varepsilon_{21} = 0.12738$ y $\varepsilon_{22} = 0.46635$.

Una alternativa al esquema de acople descrito en (8) es el empleo de un acople lineal:

$$\begin{cases} x_{n+1}^1 = \varepsilon_{11} f(x_n^1) + \varepsilon_{12} f(x_n^2) \\ x_{n+1}^2 = \varepsilon_{21} f(x_n^1) + \varepsilon_{22} f(x_n^2) \end{cases} \qquad [10]$$

La Fig. 6 muestra el conjunto de semillas que produjeron el par '11' para $\varepsilon_{11} = 0.23745$, $\varepsilon_{12} = 0.77362$, $\varepsilon_{21} = 0.12738$ y $\varepsilon_{22} = 0.46635$; se observa que esta vez cuatro iteraciones son suficientes para hacer desaparecer cualquier regularidad en el patrón observado. A la luz de estas pruebas puede concluirse que en general la correlación entre $b_n^1$ y $b_n^2$ disminuirá con el número de iteraciones y con la magnitud del acople, siendo esta disminución más pronunciada para el acople descrito por (10).

## 4. Aplicación Criptográfica

El esquema descrito por las ecuaciones (6) y (7) puede utilizarse para implementar un sistema criptográfico simétrico en el que cada grupo de bits de un texto plano binario es cifrado como el número de iteraciones necesarias para generar el mismo a partir de un conjunto de semillas. Más precisamente, sea $\{t\} = t^1,..., t^{end}$ la secuencia binaria del texto plano a cifrar, y sea $\{\alpha\} = \alpha_1,..., \alpha_L,..., \alpha_{end} = (t^1,..., t^N),..., (t^{NL+1},..., t^{N(L+1)}),..., (t^{end-N+1},..., t^{end})$ la misma secuencia $\{t\}$ pero agrupada en unidades de $N$ bits. Sea también $\{\beta\} = \beta_1,..., \beta_i,... = (b_1^1,..., b_1^N), ..., (b_i^1,..., b_i^N),...$ la secuencia de estados binarios del sistema de $N$ GCLs acoplados. Entonces se puede codificar el texto plano binario $\{t\}$ como la secuencia de posiciones $i$ que ocupan las unidades $\alpha_L$ en la secuencia $\{\beta\}$. En la práctica conviene cifrar el texto plano como la secuencia de distancias $\Delta i$ entre las posiciones de las unidades $\alpha_{L-1}$ y $\alpha_L$.

En una primera prueba esta idea se implementó para cifrar pares de bits de la representación ASCII de un texto plano utilizando 2 generadores RANDU ($2^{31}$, 65539, 0, $x_0$) idénticos con acople asimétrico $\varepsilon_{12} = 0.3$, $\varepsilon_{21} = 0.15$ y semillas $x_0^1 = 123456$, $x_0^2 = 987654$;

adicionalmente, y en vista de los resultados presentados en la sección anterior, el número de iteraciones necesarias para codificar cada par de bits es como mínimo 25 iteraciones a fines de destruir la correlación entre las secuencias individuales producidas por cada generador. A manera de ejemplo, consideremos el cifrado de la palabra 'alfa'. El texto plano binario correspondiente es $\{t\}$ = {01100001, 01101100, 01100110, 01100001}, donde por conveniencia hemos utilizado 8 bits para representar el equivalente binario del código ASCII de cada letra. Esta secuencia agrupada en pares de bits será $\{\alpha\}$ = {(0,1), (1,0), (0,0), (0,1), (0,1), (1,0), (1,1), (0,0), (0,1), (1,0), (0,1), (1,0), (0,1), (1,0), (0,0), (0,1)}. La correspondiente secuencia de estados binarios en nuestro sistema de 2 GCLs acoplados es $\{\beta\}$ = {(0,0), (1,1), (0,1), …, (1,0), (1,1), …(1,0)}. El cifrado de la palabra 'alfa' estaría dado por las posiciones ocupadas dentro de la secuencia $\{\beta\}$ por cada uno de los pares de bits en $\{\alpha\}$. Por ejemplo, el primer par de bits en la secuencia $\{\alpha\}$ es (1,0), y este par aparece después de 28 iteraciones en la secuencia $\{\beta\}$. De este modo, y recordando que las primeras 25 iteraciones de los GCLs son descartadas, la palabra "alfa" quedaría cifrada como (3, 3, 1, 13, 5, 2, 4, 4, 6, 2, 7, 4, 3, 6, 2, 13).

En una segunda prueba se emplearon dos generadores RANDU ($2^{31}$, 65539, 0, $x_0$) idénticos acoplados de acuerdo al esquema descrito por (10) con $\varepsilon_{11}$ = 0.23745, $\varepsilon_{12}$ = 0.77362, $\varepsilon_{21}$ = 0.12738, $\varepsilon_{22}$ = 0.46635 para cifrar pares de bits en la representación binaria de una imagen en escala de grises en formato PGM. Bajo este formato, la imagen es representada por una matriz cuyos elementos representan el nivel de grises de cada pixel de la imagen; a su vez, cada uno de estos elementos es un número comprendido entre 0 y 255 que puede representarse con una palabra binaria de 8 bits. La Fig. 7 muestra la imagen original, en tanto que la Fig. 8 muestra la imagen recuperada a partir del cifrado cuando hay una diferencia $\Delta\varepsilon_{11} = 10^{-16}$ en el acople $\varepsilon_{11}$. Diferencias menores a esta cantidad en el valor de los acoples no afectaron la imagen recuperada.

Teniendo en cuenta que un eventual adversario conoce el esquema de cifrado descrito anteriormente, la clave de cifrado estará definida por los parámetros $A$, $B$, $M$ de los GCL utilizados, los valores de los acoples $\varepsilon_{jk}$ y los valores de las semillas $(x_0^1, \ldots, x_0^N)$.

Figura 7: Imagen PGM original

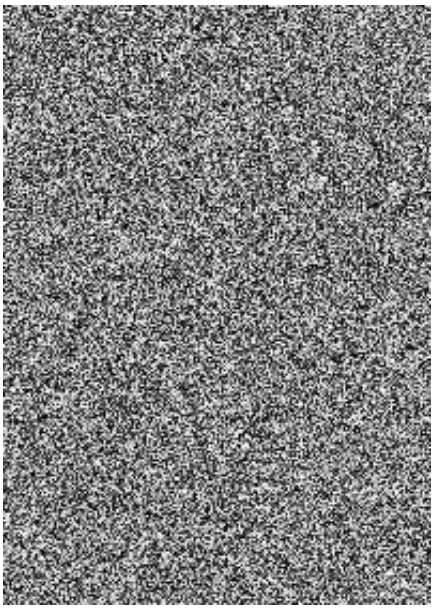

Figura 8: Imagen PGM recuperada para una diferencia $\Delta\varepsilon_{11} = 10^{-16}$

Suponiendo que se emplea el esquema de acople descrito por (10) con GCLs idénticos y el empleo de una representación binaria de 32 bits para *A*, *B*, *M,* ($x_0^1$, $x_0^2$), y teniendo en cuenta que una diferencia de $10^{-16}$ en uno de los acoples invalida la clave de cifrado, la búsqueda exhaustiva en el espacio de las posibles claves supondría el explorar $2 \times 2^{128} \times 4 \times 10^{16} \approx 10^{55}$ combinaciones. Tal como se mencionó en una sección anterior, podrían desarrollarse métodos no exhaustivos de búsqueda en el espacio de las semillas ($x_0^1$, $x_0^2$), sin embargo el desarrollo y aplicación de tales métodos presenta un grado de dificultad elevado. Por otra parte, estos métodos no serían necesariamente aplicables a la búsqueda de los parámetros *A*, *B*, *M* o de los acoples $\varepsilon_{jk}$.

Un factor adicional para hacer más difícil la búsqueda de la clave consiste en introducir una perturbación en los GCLs cada vez que un par $\alpha_i$ es cifrado. Este procedimiento hace que la secuencia $\{\beta\}$ sea dependiente del texto que está siendo cifrado. Esta técnica impide la construcción de una secuencia única $\{\beta\}$ que serviría para el descifrado de mensajes sin necesidad de realizar el algoritmo correspondiente, y por lo tanto sin necesidad de conocer la clave [6].

Tal como se deduce de la Fig. 6, el empleo del esquema de acople descrito por (10) hace que la correlación entre las secuencias ($b_n^1$, $b_n^2$) sea despreciable tras un número pequeño de iteraciones. Este hecho, aunado a la rapidez computacional de un GCL, sugiere que un esquema criptográfico basado en (10) podría emplearse para cifrado de audio en tiempo real o cuasi real.

## 5. CONCLUSIONES

Se ha presentado algunas características de un sencillo generador de números pseudo aleatorios de congruencia lineal (GCL), enfatizando aquellas propiedades que lo hacen inapropiado para aplicaciones criptográficas. Se presentó una técnica de cifrado simétrico computacionalmente segura basada en el colapso del espacio de fases de un conjunto de *N* GCLs idénticos acoplados, la cual permite cifrar grupos de *N* bits en la representación binaria del texto plano original; en estas condiciones la clave de cifrado estará constituida por los valores de los parámetros (*A*, *B*, *M*) del GCL, las semillas ($x_0^1$, ..., $x_0^N$), y los coeficientes de acople. La determinación de esta clave por medios de búsqueda exhaustiva es computacionalmente inaccesible. El esquema criptográfico propuesto es rápido y de fácil implementación, teniendo como principal desventaja el hecho de que el tamaño del texto cifrado es mayor que el del texto plano original.

**Referencias Bibliográficas**


1. LEHMER D.H. Mathematical Methods in Large-Scale Computing Units. *2nd Symposium On Large-Scale Digital Calculating Machinery*, Cambridge, Massachussets (USA), 141-146, 1949.
2. GREENBERGER M. **Journal of the ACM**, Vol. 8, Issue 2, 163-167, 1961.
3. MARSAGLIA G. **Proceedings of the National Academy of Sciences of the USA**, 61(1), 25-28, 1968.
4. BELLARE M, GOLDWASSER S, MICCIANCIO D. Pseudo-Random" Number Generation within Cryptographic Algorithms: The DDS Case. *Advances in Cryptology – Crypto 97 Proceedings, Lecture Notes in Computer Science*. Santa Bárbara, California (USA), Vol. 1294, pp. 287-291, 1997.
5. BOYAR J. **Journal of the ACM**, 36(1), 129-141, 1989.
6. GARCIA P, PARRAVANO A, COSENZA M, JIMENEZ J, MARCANO A. **Phys Rev E**, Vol. 65 (4) 045201(R), 2002.
7. WANG S, LIU W, LU H, KUANG J, HU G. nlin/0309005.
8. SCHUSTER H.G. "**Deterministic Chaos: An Introduction**" 3ra edición. VCH Verlagsgesellschaft, Weinheim (Germany), pp. 21-23, 1995.