Recibido 11 Sep 2016 *ReCIBE, Año 6 No. 1, Mayo 2017*
Aceptado 15 Mar 2017

Debilidad SAC en el algoritmo de cifrado en flujo RC4

SAC Weakness in RC4 stream cipher

Evaristo J. Madarro Capó[1]
evaristoj@uclv.cu

Oristela Justiz Cuellar[1]
oristela@uclv.edu.cu

Carlos M. Legón[2]
clegon@ceis.cujae.edu.cu

Guillermo Sosa Gómez[3]
guillermo.sosa@cimat.mx

[1] UCLV, Departamento de Matemática
Universidad Central "Marta Abreu" de Las Villas
Santa Clara, Cuba
[2] UH, Instituto de Criptografía. CUJAE. MATCOM.
Universidad de la Habana, Cuba.
[3] CIMAT, Centro de Investigaciones en Matemáticas, A.C.,
Guanajuato, México

Resumen: En este trabajo se describe una interesante vía para la detección de la existencia de entradas de colisión en el algoritmo de cifrado en flujo RC4 basado en el criterio estricto de avalancha y se discuten los resultados obtenidos en comparación con las debilidades y ataques reportados sobre el RC4 que plantean la existencia de una fuerte correlación entre los parámetros de entrada y salida del algoritmo.

Palabras Clave: criterio estricto de avalancha, entradas de colisión, correlación estadística, RC4.

Abstract: This paper describes an interesting way for the detection of the colliding entries existence in the RC4 stream cipher algorithm based in the strict avalanche criterion and the results obtained are discussed in comparison with the reported weaknesses and attacks on the RC4 that raise the existence of a strong correlation between the parameters of Input and output of the algorithm.

Keywords: strict avalanche criterion, colliding entries, statistical correlation, RC4.

1. Introducción

El desarrollo continuo de las redes de comunicación, en particular de la Internet, abre nuevas posibilidades para el intercambio de información. Al mismo tiempo, son cada vez mayores las amenazas a la seguridad de la información que se transmite, por lo que ha sido necesario entonces, la creación de diferentes mecanismos, dirigidos a garantizar la integridad, confidencialidad y autenticidad de los documentos electrónicos.

Entre las herramientas más utilizadas se encuentra la criptografía. Actualmente existen propuestas muy interesantes de algoritmos de cifrado (Schneier, 1996) para garantizar la confidencialidad de la información y entre estos se encuentran los algoritmos de cifrado en flujo. El algoritmo RC4 (Mantin, 2001) (Paul & Maitra, 2012) se ha destacado, particularmente, por su amplio uso en diferentes protocolos y aplicaciones (*SSL, WEP*) (Fluhrer, Mantin, & Shamir, 2002).
A través de los años este algoritmo ha sido extensamente evaluado y analizado en la literatura, encontrando varias propiedades interesantes y algunas debilidades en el proceso de la inicialización. Algunas de estas expresan que existen entradas distintas que pueden generar sucesiones de salida correlacionadas. Esta propiedad es llamada colisiones entre entradas (Wallach & Grosul, 2000) (Matsui, 2009) (Chen & Miyaji, 2011).

En este trabajo se describen estos criterios y algunas valoraciones estadísticas publicadas acerca de las debilidades del RC4 y se confirman en la práctica mediante la propuesta de una interesante vía para la detección de la correlación existente entre los parámetros de entrada y salida del RC4.

2. Descripción el algoritmo RC4

El algoritmo RC4 opera en palabras binarias de longitud n, en la práctica n = 8 (Mantin, 2001) (Paul & Maitra, 2012). En cada iteración el RC4 produce una salida que puede tomar cualquiera de los $N = 2^n$ posibles valores. Para n = 8, el algoritmo consta de una permutación S del conjunto {0,…, 255} y dos variables i y j de un byte cada una, utilizadas como indicadores a los elementos de S. Inicialmente, las dos variables, i y j, se inicializan en cero y la permutación en la identidad (Mantin, 2001) (Paul & Maitra, 2012).

El funcionamiento del RC4 se divide en dos algoritmos (Mantin, 2001) (Paul & Maitra, 2012), tal y como se ilustra en la Fig. 1:
- El algoritmo de esquema de llaves (Key Scheduling Algorithm, KSA), el cual construye una permutación S a partir de una entrada K (llave) y la permutación inicial.

- El algoritmo generador pseudoaleatorio (Pseudo Random Generator Algorithm, PRGA), utilizado para generar como salidas sucesiones de bytes de longitud deseada.

```
for i ← 0 to 255 do
  S[i] ← i
end
j ← 0
for i ← 0 to 255 do
  j ← j+S[i]+K[i mod len(K)] mod 256
  swap(S, i, j)
end
i ← 0
j ← 0
```

```
i ← i + 1 mod 256
j ← j + S[i] mod 256
swap(S, i, j)
return S[ S[i] + S[j] mod 256 ]
```

Figura 1. *KSA y PRGA*

Con cada byte de salida producido por el PRGA, el estado interno del RC4 es actualizado.
Al comparar KSA y PRGA se observa que la única diferencia está en la manera de incrementar la variable j, añadiéndole un byte de la entrada en cada iteración, teniendo en cuenta la longitud de la entrada.

El KSA genera una permutación de inicio (estado inicial) para el PRGA a partir del parámetro de entrada K de k bytes y la permutación inicial (identidad). Una vez que el estado inicial es obtenido, este es utilizado por el PRGA. El propósito del PRGA es generar la sucesión de salida, en este caso en bytes.
Su implementación es extremadamente sencilla y rápida, y está orientado a generar secuencias en unidades de un byte, además de permitir entradas de diferentes longitudes

3. Debilidades del algoritmo RC4

Entre las debilidades encontradas al algoritmo RC4 están las colisiones entre entradas relacionadas de tal manera que la distancia de Hamming entre ambas es uno, introducidas por Grosul y Wallach (Wallach & Grosul, 2000), luego ampliado por Matsui (Matsui, 2009) y generalizado en (Chen & Miyaji, 2011).

Matsui (Matsui, 2009) estudia las denominadas "entradas de colisión" del RC4 que crean el mismo estado inicial y generan el mismo flujo de byte pseudoaleatorio. Su principal aporte en este trabajo es motivado por el hecho de que hasta ese momento (2009) solo era observable la existencia de entradas de colisión cuando el tamaño de la entrada en el algoritmo RC4 era muy *grande* (Matsui, 2009), es decir, se desconocía la existencia de colisiones entre entradas para tamaños *cortos*. Los pares de entradas de colisión para tamaños de entrada *grande* fueron expuestos en el 2000 por Grosul y Wallach

(Wallach & Grosul, 2000) al obtener para tamaños de entrada cercanos a los 256 bytes salidas de cientos de bytes similares sustancialmente.

El número total de posibles estados iniciales del RC4 es $256! \approx 2^{1684}$ y debido a esto Matsui (Matsui, 2009) plantea que, en el RC4, deben existir entradas de colisión si el tamaño de la entrada excede los $\lfloor 1684/8 \rfloor = 210$ bytes.
No obstante, obtiene como resultado que un par de entradas relacionadas de longitud fija arbitraria y menor tamaño puede conducir a entradas de colisión y muestra como ejemplo un par de entradas de colisión de 24 bytes. También demuestra que es muy probable que el RC4 tenga un par de entradas de colisión incluso si el tamaño de entrada es menor de 20 bytes y propone un algoritmo recursivo para la búsqueda de entradas de colisión.

Luego, Chen y Miyaji en (Chen & Miyaji, 2011) realizan una generalización de los resultados de Matsui a través de los siguiente puntos:

1. Muestran que el RC4 puede generar pares de entradas de colisión para varias distancias de Hamming, los cuales no pueden ser generados por el patrón de Matsui, y algunos ejemplos concretos de pares de entradas de colisión con distancia de Hamming mayor que 1.
2. Formalizan los pares de entradas de colisión en el RC4 en dos amplios patrones de entradas de colisión, a los cuales denominan patrón *Transitional* y patrón *Self-Absorbing*, y plantean que todos los pares de entradas de colisión conocidos actualmente pueden ser categorizados en uno de estos dos patrones.
3. Analizan y explican las relaciones entre la probabilidad de una entrada de colisión, la longitud de la entrada y la distancia de Hamming, que proporcionan pares de entradas de colisión.

4. Detección de correlación en el RC4

En este trabajo se utiliza una prueba estadística propuesta en (M. Capó, Legón, Cuellar, & Sosa, 2016), basada en el criterio estricto de avalancha (*strict avalanche criterion*, SAC) (Adams & Tavares, 1990), para medir el nivel de correlación existente entre las salidas del algoritmo RC4 correspondientes a entradas que solo difieren en un bit.

Un algoritmo de cifrado en flujo es la especificación de un algoritmo de generación de números pseudoaleatorios el cual se puede asociar a una función $f: \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, la cual recibe n bits de entrada y tiene m bits de salida. En caso de que la salida del generador no sea de tamaño m se iterará hasta alcanzar dicha longitud de salida.

Esta prueba está basada en el cambio de un bit en cada uno de los índices de una entrada prefijada y aplicar el criterio estricto de avalancha entre la entrada

original y cada una de las obtenidas al variar un bit para medir la correlación entre sus respectivas salidas. Es decir, asociar la existencia de entradas de colisión a la cantidad de componentes de las salidas que varían al alterar los índices y que provocan salidas correlacionadas.

Contrariamente a la definición del SAC (Adams & Tavares, 1990), no es posible aplicar esta prueba para todas las posibles entradas de manera que solo se aplica a un subconjunto $X = \{x_0, \dots, x_h\}$ de h de entradas escogidas aleatoriamente. Luego, se evalúa cada uno de las entradas x_i de n bits en la función, para obtener una sucesión binaria de salida $s_i = f(x_i)$ de m bits. Después se evalúan en esta función las entradas x_j de n bits, con $x_j = (x_i \oplus C_j^n)$, con $1 \leq i \leq h$ y $1 \leq j \leq n$, y se determina la distancia de Hamming, por componentes, entre la sucesión obtenida $s_j = f(x_j)$ y la sucesión anterior $s_i = f(x_i)$, para cada $1 \leq j \leq n$. Cuando se habla de calcular la distancia de Hamming por componentes se refiere a realizar la operación $\sum_{r=1}^{k}(s_i \oplus s_j)$, donde $(\sum)$ se realiza en $\mathbb{F}_2^m$. El procedimiento finaliza al medir el ajuste de la distribución observada a la distribución $B(\frac{1}{2}, k)$, mediante la prueba de bondad de ajuste chi-cuadrado.

Si el criterio se satisface para cada una de las sucesiones binarias generadas compuestas de k entradas, entonces se asume que el generador satisface el criterio estricto de avalancha.

La prueba estadística propuesta consiste en interpretar a $f(x)$ como la salida del RC4 y está basada en el algoritmo presentado en (M. Capó, Legón, Cuellar, & Sosa, 2016), en pseudocódigo, para detectar la correlación entre las entradas y salidas de un PRNG en general. Este algoritmo se resume en los siguientes pasos:

Algoritmo PRNG-SAC

Entrada: Subconjunto $X = \{x_0, \dots, x_h\}$ de h entradas escogidas aleatoriamente

Salida: Si satisface o no el SAC;

1. Para cada una de las entradas x_i de n bits realizar lo siguiente
   1.1. Generar las sucesiones de salida de m bits $s_i = f(x_i)$ y $s_j = f(x_i \oplus C_i^n)$
2. Computar $\Sigma_{i,j}(s_i \oplus s_j)$
3. Aplicar prueba chi cuadrado para determinar el ajuste al criterio.

Si el algoritmo satisface el criterio SAC entonces todas las sucesiones s_j son independientes de s_i, no tienen correlación con s_i, por tanto $P(s_i = s_j) = \frac{1}{2}$, con $1 \leq i \leq h$ y $1 \leq j \leq n$, y

$$\sum_{r=1}^{k}(s_i \oplus s_j) \sim B(\frac{1}{2}, k)$$

Para satisfacer el SAC, a través de la prueba de bondad de ajuste chi cuadrado, se requiere contrastar las hipótesis:

$$H_0: p = \frac{1}{2} \qquad H_1: p \neq \frac{1}{2}$$

El estadígrafo utilizado es

$$z_w = \frac{\left(n_1^w - \frac{k}{2}\right)^2}{\frac{k}{4}}$$

donde n_1^w es el valor del w-ésimo componente y $n_0^w = k - n_1$, $1 \leq w \leq m$. Como existen solo dos categorías posibles, se tiene 1 grado de libertad.

Luego, si $z_w \leq z_\alpha$ se acepta la hipótesis nula H_0. Así se tiene que

$$P(z_w \leq z_\alpha) = 1 - \alpha$$

y

$$P(z_w > z_\alpha) = \alpha$$

Como se tienen m observaciones el número esperado de fallos, número de veces se espera se rechace H_0, es $\alpha \cdot m$.

5. Análisis Experimental

Para la realización de los experimentos se implementó el algoritmo referenciado para dos pares de parámetros (n, m), tomando (n = 20, m = 20) y (n = 256, m = 256) respectivamente, para constatar los resultados referenciados.

Al aplicar la prueba al algoritmo criptográfico RC4 en el caso de n = 20 y m =20 se observó que en la mayoría de los casos se acepta la hipótesis nula como se muestra en la Tabla 1.

Con un nivel de significancia α = 0,01 el número esperado $E(n_1)$ de componentes para los cuales se rechaza la hipótesis nula es aproximadamente $E(n_1)$ = 1.60 para cada uno de los 20·8=160 cambios en un bit. En este caso se escogió aceptar la hipótesis nula si $n_1 \leq 2$.

Intervalo	$0 \leq n_1 \leq 2$	$2 < n_1 \leq 3$	$3 < n_1 \leq 4$	$4 < n_1 \leq 5$
Cantidad de cambios fallidos	113	27	12	7

Tabla 1. *Distribución por intervalos de los resultados para n =20 y m =20*

Esto determina que en promedio este algoritmo posee buena confusión para estos parámetros. Además, se puede notar que si las entradas son seleccionadas al azar no es fácil obtener pares que cumplan las características de (Matsui, 2009) (Chen & Miyaji, 2011).

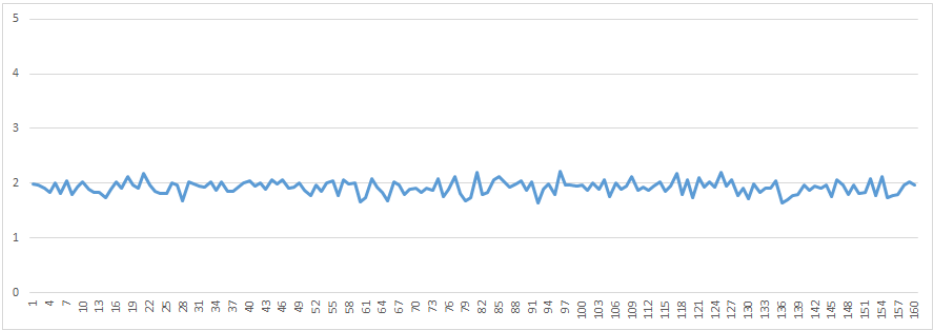

Figura 2. *Distribución del número de componentes rechazados por cada uno de los 160 cambios de bit*

No obstante, para los parámetros n = 256 y m = 256 se obtuvo que en la mayoría de los casos se rechaza la hipótesis nula como se muestra en la Tabla 2.

Con un nivel de significancia $\alpha = 0.01$ el número esperado $E(n_1)$ de componentes para los cuales se rechaza la hipótesis nula es aproximadamente $E(n_1) = 20.48$ para cada uno de los 256·8 = 2048 cambios en un bit. Aquí se decidió aceptar la hipótesis nula si $n_1 \leq 21$.

Por ejemplo, solo en 412 de los cambios se obtuvo que 21 o menos componentes no satisfacen la hipótesis nula.

Intervalo	$0 \leq n_1 \leq 21$	$21 < n_1 \leq 30$	$31 < n_1 \leq 100$	$n_1 > 100$
Cantidad de cambios fallidos	412	1042	465	129

Tabla 2. *Distribución por intervalos de los resultados para $n = 256$ y $m = 256$*

Es realmente difícil ilustrar la cantidad de componente rechazados en cada uno de los cambios de bit debido al volumen de trabajo. No obstante, mediante el siguiente gráfico es posible tener una panorámica sobre cómo se comportó en cada uno de los cambios el número de fallos.

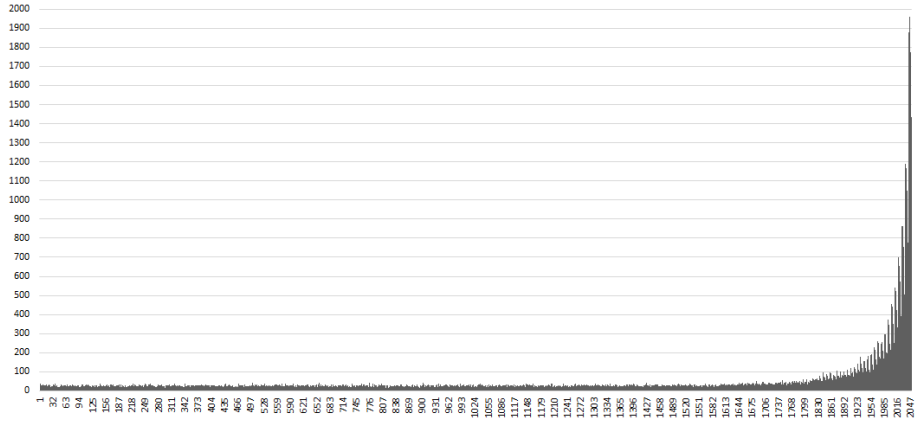

Figura 3. *Distribución del número de componentes rechazados por cada uno de los 2048 cambios de bit.*

Algo a resaltar, de los resultados mostrados en la figura anterior, es el rechazo de la hipótesis nula que a partir del cambio en el bit 1800 (byte 225 de la entrada) en adelante, llegando a tomar valores desfavorables en el cambio de los últimos 48 bits. En la siguiente tabla se muestra el comportamiento en los últimos 16 cambios de bit.

Bit cambiado	Cantidad de Fallos	Bit cambiado	Cantidad de Fallos
2033	1163	**2041**	1877
2034	1188	**2042**	1838
2035	1167	**2043**	1850
2036	1140	**2044**	1960
2037	1051	**2045**	1776
2038	1002	**2046**	1570
2039	779	**2047**	1551
2040	677	**2048**	1436

Tabla 3. *Distribución de fallos en los ultimos 16 bits*

Los resultados alcanzados constatan lo obtenido anteriormente en (Wallach & Grosul, 2000) (Matsui, 2009) (Chen & Miyaji, 2011). Es decir, para entradas $K1[i] = K2[i]$ que se diferencian solo en una posición t y dicha diferencia es 1 ($K1[t] = K2[t] - 1$), cuando *t* se acerca a la longitud de K hay un mayor chance de que exista una alta correlación entre las sucesiones de salida.

De esta manera, esta prueba estadística permite la detección de la existencia de entradas de colisión en el algoritmo de cifrado en flujo RC4, midiendo el nivel de correlación existente entre sus salidas.

6. Conclusiones y Trabajo Futuro

En este trabajo se expusieron los resultados de aplicar al algoritmo de cifrado en flujo RC4 una prueba estadística propuesto en (M. Capó, Legón, Cuellar, & Sosa, 2016) para determinar la dependencia entre las sucesiones de salida de generadores de números pseudoaleatorios y los valores de entrada determinados de forma tal que solo se diferencian en un bit.
La prueba propuesta es capaz de detectar la presencia de anomalías en la dependencia estadística entre las entradas y las salidas del RC4 tal y como se observa en la Fig. 3. incluso para muestras pequeñas.

Los resultados prácticos alcanzados corroboran la dificultad de hallar entradas de colisión para entradas de longitud 20. Mientras para entradas con una longitud cercana a la longitud máxima 256 es posible encontrar sin mucho esfuerzo computacional este tipo de entradas.

Como próximo trabajo en esta línea los autores se proponen generalizar el método de manera que se pueda evaluar la dependencia estadística con respecto a entradas con más de un bit de diferencia. Además, el estudio de los valores óptimos de los parámetros (n, m), que determinan el tamaño de muestra, para obtener un resultado más preciso, que se ajuste mejor a la calidad real de los algoritmos a evaluar. Estos parámetros pueden determinar el buen rendimiento y la rigurosidad de esta prueba, como ilustran los resultados obtenidos en (Tsang, Hui, Chow, & Chong) para la conocida prueba de colisiones propuesta por Knuth (Knuth, 1985).

Referencias:

Adams, C., & Tavares, S. (1990). The Structured Design of Cryptographically Good S-boxes.

Chen, J., & Miyaji, A. (2011). Generalized Analysis on Key Collisions of Stream Ciphers RC4.

Fluhrer, S., Mantin, I., & Shamir, A. (2002). Attacks on RC4 and WEP.

Knuth, D. (1985). The Art of Computer Programming (Volume 2).

M. Capó, E., Legón, C., Cuellar, O., & Sosa, G. (2016). Evaluation of Input - Output Statistical Dependence PRNGs by SAC. International Conference on Software Process Improvement (CIMPS). IEEE Digital Library.

Mantin, I. (2001). Analysis of the stream ciphers RC4.

Matsui, M. (2009). Key Collisions of the RC4 Stream Cipher.

Paul, G., & Maitra, S. (2012). RC4 Stream Ciphers and its Variants. Discrete Mathematics and Its Applications.

Schneier, B. (1996). Applied Cryptography.

Tsang, W., Hui, L., Chow, K., & Chong, C. (n.d.). Tuning the Collision Test for Stringency.

Wallach, D., & Grosul, A. (2000). A Related-Key Cryptanalysis of RC4. Rice University: Technical Report TR-00-358, Department of Computer Science.

Notas biográficas:

Evaristo J. Madarro Capó es master en ciencia de la computación por la Universidad Central “Marta Abreu” de las Villas, Villa Clara. Previamente, en el Instituto Superior Politecnico José Antonio Echeverría, La Habana, cursó una especialidad en Ingienería en Aplicaciones Criptograficas y la ingeniería en ciencias informáticas en la Universidad de Ciencias Informáticas (UCI), La Habana. Actualmente labora como Prof. Asistente en el grupo de investigación del Instituto de Criptografía, Universidad de la Habana, La Habana, Cuba, donde participa en proyectos nacionales de investigación con entidades educativas. Como investigador, sus áreas de interés son: matemática computacional y criptografia.

Oristela Cuellar Justiz es Doctora en Ciencias Matemáticas en enero del 2017 en la especialidad Matemática Computacional titulada "Homomorfismos de inmersión y matrices MDS en la Criptografía". Máster en Matemática Aplicada em 2007 y Licenciada en Matemática y Física en julio de 1987 en el Instituto Pedagógico Estatal. León Tolstoi. Tula. URSS. Es profesora de matemática en la UCLV desde agosto del 2001. Ha impartido en los últimos cinco años en pregrado las siguientes asignaturas: Algebra Lineal y Teoría de Grupos en la carrera de Licenciatura en Química, Optimización Matemática I y II, Algebra III, Optativa Campos finitos y sus aplicaciones en la carrera de Licenciatura en Matemática y Matemática III en la carrera de Informática 2do año. En postgrado impartió el curso "Campos finitos y sus aplicaciones en la Criptografía". Ha participado en los proyectos de investigación: Investigación y diseño de algoritmos criptográficos de cifrado en flujo, Criptología y los algoritmos de cifrado en flujo, Las estructuras algebraicas y los algoritmos de cifrado en flujo que se inició en febrero del 2017. Dirige el Laboratorio de Criptografía Académica de la Universidad y el grupo de Algebra y Criptografía. Recibió el Premio Anual al Mérito Científico en la categoría de Resultado de mayor aporte contribución a la Defensa y Seguridad Nacional en el año 2013. (Coautora) y el Premio CITMA provincial 2014. ”Solución de problemas sobre campos finitos". (como autora principal). Tiene un registro de software como coautora: Software:" para la solución de problemas sobre campos finitos binarios. BiGFSoP. Número de registro 2741-09-2014. Sus publicaciones más importantes de los últimos seis años han sido en las Revistas : " Chebyshevsky Sbornik " de Rusia; Ciencias Matemáticas de Cuba; Journal of Advances in Mathematics y Proceeding of the 5th International Conference in Software Process Improvement. CIMPS 2016.

Carlos M. Legón Pérez es graduado de Licenciatura en Matemáticas en 1981 y Doctor en Ciencias Matemáticas en 1996. Profesor Titular en 2000 e Investigador Titular en 1998. A partir de su graduación en 1981, se desempeñó como profesor universitario e investigador científico, en un centro de seguridad informática del Ministerio de la Informática y las Comunicaciones (MIC), trabajando en este lugar hasta abril del 2009. Profesor Titular de la Facultad de Ingeniería Informática de la CUJAE desde 2010 hasta 2017. Investigador Titular del Instituto de Criptografía de la Facultad de Matemática y Computación de la Universidad de la Habana desde 2010 hasta 2017. Sus temas de investigación han estado orientados a la aplicación de modelos y métodos matemáticos para el planteamiento y solución de problemas técnicos surgidos en la informática y las comunicaciones, de forma particular en la seguridad informática y específicamente en el campo de la criptografía, probabilidades y estadísticas - teoría de información aplicada a la criptografía, ataques de canal colateral.

Guillermo Sosa Gómez con dirección particular: Villa San Luis de la Paz # 18. Villas de Guanajuato, Guanajuato. Código postal: 36250 y teléfono: 4735978324. Situación profesional precedente: Ministerio de Educación Superior. Facultad: Matemática, Física y Computación, Cuba. Categoría profesional Docente: Profesor Auxiliar con cargos: Segundo Jefe del Departamento de Matemática y Coordinador de la Carrera de Licenciatura en Matemática y experiencia laboral: 14 años. Mi formación académica: Titulación: Licenciado en Matemática (Título de Oro) Centro: Universidad Central de Las Villas en julio de 2007 y Maestría en Matemática Centro: Universidad Central de Las Villas Fecha en julio de 2010. Cursos de postgrado recibidos: Metodología de la investigación, Simulación y Métodos de Monte Carlo, Estadística Multivariada, Programación Dinámica, Diseño Estadístico Experimental, Introducción a la Criptografía Cifrado en Flujo, Funciones hash Curvas Elípticas, Estadística y probabilidades aplicadas a la Criptografía, Problemas sociales de la Ciencia y la Tecnología, Formación Pedagógica, Matemática Computacional, Ecuaciones Diferenciales y Calculo Variacional. Situación profesional actual: Alumno de Doctorado en Matemática Básica en CIMAT, A.C. donde su tema está dedicado a la construcción de funciones booleanas con buenas propiedades criptográficas