Ciencias Matemáticas, Vol. 35, No. 1, 2021, Pag. 47-53
Recibido 11-2020, Aceptado 04-2021

Análisis y diseño de variantes del criptoanálisis a cifrados en bloques mediante el Algoritmo Genético
Analysis and design of cryptanalysis variants to block ciphers through Genetic Algorithm

Osmani Tito Corrioso[1]*, Miguel A. Borges Trenard[2], Mijail Borges Quintana[3]

Resumen En los últimos años es cada vez más creciente el uso que se le ha dado al Algoritmo Genético (AG) en el criptoanálisis a cifrados en bloques. No obstante, todavía es necesario seguir profundizando en su estudio y la búsqueda de mejores prestaciones. En ese sentido, en este trabajo se realiza el estudio de ciertas variantes y parámetros relacionados con el algoritmo, como: las funciones de aptitud, el valor de k_2 (que determina un balance entre la cantidad de clases en que se divide el espacio de las claves y la cantidad de elementos de cada una de ellas), y una metodología de ataque donde se logran desechar algunas clases, reduciendo el número total que sería necesario recorrer. Los experimentos se realizan con los cifrados HTC, AES(3), AES(4) y AES(7).

Abstract In recent years, the use that has been given to the Genetic Algorithm (GA) in the cryptanalysis of block ciphers has been increasing. However, it is still necessary to continue deepening its study and the search for better benefits. In this sense, in this work the study of certain variants and parameters related to the algorithm is carried out, such as: fitness functions, the value of k_2 (which determines a balance between the number of classes into which the space of the keys and the number of elements of each one of them), and an attack methodology where some classes are discarded, reducing the total number that would be necessary to go through. The experiments are carried out with the HTC, AES (3), AES (4) and AES (7) ciphers.

Palabras Clave
Algoritmo Genético, grupo cociente, criptoanálisis, AES(t)

Keywords
Genetic Algorithm, quotient group, cryptanalysis, AES(t)

[1] *Departamento de Matemática, Facultad de Ciencias de la Educación, Universidad de Guantánamo, Cuba, osmanitc@cug.co.cu*
[2,3] *Departamento de Matemática, Facultad de Ciencias Naturales y Exactas, Universidad de Oriente, Cuba, [2]mborges@uo.edu.cu, [3]mijail@uo.edu.cu*
***Autor para Correspondencia, Corresponding Author**

Introducción

El Algoritmo Genético (AG) es un método de optimización utilizado en los últimos años en la Criptografía con diversos propósitos, en particular con el de realizar ataques a varios tipos de cifrados. Algunas de las investigaciones realizadas en esta dirección se mencionan abajo.

En [14] los autores presentan una combinación del Algoritmo Genético con la Optimización de Enjambres de Partículas (otro método heurístico basado en técnicas evolutivas), llamaron a su método "Optimización de Enjambre Genético" y lo aplicaron para atacar el DES. Sus resultados experimentales muestran que se obtienen mejores resultados aplicando su método combinado que utilizando ambos métodos por separado. [6] proporciona una exploración preliminar del uso del AG sobre un cifrado del tipo Red de Sustitución Permutación (SPN por sus siglas en inglés). El propósito de la exploración es determinar cómo encontrar claves débiles. Ambos trabajos ([14] y [6]) usan un ataque a texto claro conocido, es decir, dado un texto claro T y el correspondiente texto cifrado C, se está interesado en encontrar la clave K. En [6], la función de aptitud evalúa la diferencia bit a bit (distancia de Hamming) entre C y el texto cifrado de T, usando un candidato para la clave, mientras que, por el contrario, en [14] se mide la distancia de Hamming entre T y el descifrado del texto cifrado de C. En [8] se muestra un ataque sólo a texto cifrado al SDES, obteniendo mejores resultados que por fuerza bruta. Los autores usan una función de aptitud que es una combinación de la frecuencia relativa de monogramas, digramas, y trigramas (para un idioma particular). Como la longitud de clave es muy pequeña, pudieron usar este tipo de función. [1] es similar a [8], se utiliza en esencia la misma función de aptitud, pero con diferentes parámetros, también es más detallado sobre los experimentos y los comparan no sólo con respecto a la fuerza bruta, sino también con la búsqueda aleatoria. Para más detalles sobre el área del criptoanálisis mediante el Algoritmo Genético ver [7], [2] y [10].

Aquí se sigue la idea de [4] y [11], y sus metodologías para dividir el espacio de las claves. Así como en [13], donde se propone una solución al Problema Probabilístico de Pertenencia y a la elección de k_2 (ver además [12]), temas que aquí se comprueban experimentalmente.

En el trabajo, se estudian varios parámetros del criptoanálisis a cifrados en bloque mediante el AG. Se comprueba el valor que podría tomar k_2 (que determina un balance entre la cantidad de clases en que se divide el espacio de las claves y la cantidad de elementos de cada una de ellas), y una metodología de ataque donde se logran desechar algunas clases, reduciendo el número total que sería necesario recorrer. Teniendo en cuenta este último punto, se propone el uso de otras funciones de aptitud.

La estructura del trabajo es como sigue. En la sección 1 se explican las dos metodologías de partición del espacio de las claves y el problema de la pertenencia de las claves a clases de equivalencia. En la 2 se presentan los resultados del estudio de las funciones de aptitud (subsección 2.1) y del valor de k_2 y la cantidad de clases de equivalencia (subsección 2.2). Las conclusiones en la sección 3 y por último las referencias bibliográficas.

1. Preliminares

1.1 Metodologías de partición del espacio de las claves

Sea $\mathbb{F}_2^{k_1}$ el espacio de las claves de longitud $k_1 \in \mathbb{Z}_{>0}$. Es conocido que $\mathbb{F}_2^{k_1}$ tiene cardinal 2^{k_1} y por tanto hay una correspondencia uno a uno entre $\mathbb{F}_2^{k_1}$ y el intervalo $[0, 2^{k_1}-1]$. Si se fija un entero k_2, $(1 < k_2 \leq k_1)$, entonces el espacio de las claves puede ser representado por los números,

$$q2^{k_2} + r, \tag{1}$$

donde $q \in [0, 2^{k_1-k_2}-1]$ y $r \in [0, 2^{k_2}-1]$. De esta forma el espacio de las claves queda dividido en $2^{k_1-k_2}$ bloques (determinados por el cociente en el algoritmo de la división dividiendo por 2^{k_2}) y, dentro de cada bloque, la clave correspondiente está determinada por su posición en el bloque, la cual está dada por el resto r. La idea principal es situarse en un bloque (dado por q) y moverse dentro de dicho bloque por los elementos (dados por r) usando el AG. Notar en esta metodología que primero se fija q para escoger un bloque y luego r varía para poder moverse por los elementos del bloque, pero la clave completa en $\mathbb{F}_2^{k_1}$ se obtiene por la fórmula (1). Haremos referencia a esta metodología como BBM. Para más detalle de la conexión con el AG ver [4].

La siguiente metodología se basa en la definición y cálculo del grupo cociente de las claves G_K cuyo objetivo es hacer una partición de $\mathbb{F}_2^{k_1}$ en clases de equivalencia. Es conocido que $\mathbb{F}_2^{k_1}$ como grupo aditivo, es isomorfo a $\mathbb{Z}_{2^{k_1}}$. Sea h el homomorfismo definido del modo siguiente:

$$\begin{aligned} h : \mathbb{Z}_{2^{k_1}} &\longrightarrow \mathbb{Z}_{2^{k_2}} \\ n &\longrightarrow n \,(\mathrm{mod}\, 2^{k_2}), \end{aligned} \tag{2}$$

donde $k_2 \in \mathbb{Z}_{>0}$ y $0 < k_2 < k_1$. Denotemos por N al núcleo de h, es decir,

$$N = \{x \in \mathbb{Z}_{2^{k_1}} | h(x) = 0 \in \mathbb{Z}_{2^{k_2}}\}. \tag{3}$$

Luego, por la definición de h se tiene que N está formado por los elementos de $\mathbb{Z}_{2^{k_1}}$ que son múltiplos de 2^{k_2}. Se sabe que N es un subgrupo invariante (o normal), por tanto, el principal objetivo es calcular el grupo cociente de $\mathbb{Z}_{2^{k_1}}$ por N y de esta forma el espacio de las claves quedará dividido en 2^{k_2} clases de equivalencia.

Denotemos por G_K al grupo cociente de $\mathbb{Z}_{2^{k_1}}$ por N ($G_K = \mathbb{Z}_{2^{k_1}}/N$). Por el Teorema de Lagrange, se tiene que $o(G_K) = o(\mathbb{Z}_{2^{k_1}})/o(N)$, pero $o(G_K) = o(\mathbb{Z}_{2^{k_2}}) = 2^{k_2}$, luego,

$$o(N) = o(\mathbb{Z}_{2^{k_1}})/o(\mathbb{Z}_{2^{k_2}}) = 2^{k_1-k_2}. \tag{4}$$

Ahora se puede describir a N, teniendo en cuenta que sus elementos son los múltiplos de 2^{k_2}. Para ello tomemos $Q = \{0, 1, 2, \ldots, 2^{k_1-k_2}-1\}$, entonces:

$$\begin{aligned} N &= <2^{k_2}> = \{x \in \mathbb{Z}_{2^{k_1}} | \, \exists\, q \in Q, x = q2^{k_2}\} = \\ &= \{0, 2^{k_2}, 2*2^{k_2}, 3*2^{k_2}, \ldots, (2^{k_1-k_2}-1)*2^{k_2}\}. \end{aligned} \tag{5}$$

Por otra parte,

$$G_K = \{N, 1+N, 2+N, \ldots, (2^{k_2}-2)+N, (2^{k_2}-1)+N\}.$$

De esta forma $\mathbb{Z}_{2^{k_1}}$ queda dividido en una partición de 2^{k_2} clases dadas por N. A G_K se le denomina *grupo cociente de las claves*.

Sea,

$$E : \{0,1\}^m \times \{0,1\}^n \to 0, 1^n, m, n \in \mathbb{N}, m \geq n, \tag{6}$$

un cifrado en bloques, T un texto plano, K una clave y C el correspondiente texto cifrado, o sea, $C = E(K, T)$; se dice que K' es una *clave consistente* con E, T y C, si $C = E(K', T)$ (ver [4]). La idea aquí también es recorrer, del espacio total, los elementos que se encuentran en una clase, para luego encontrar una (o varias) *claves consistentes* en esa clase. Para poder recorrer los elementos de cada clase, notar que $\mathbb{Z}_{2^{k_2}}$ es isomorfo con G_K y el isomorfismo hace corresponder a cada $r \in \mathbb{Z}_{2^{k_2}}$ su clase de equivalencia $r+N$ en G_K, se tiene que, seleccionar una clase es fijar un elemento $r \in \mathbb{Z}_{2^{k_2}}$. Por otro lado, los elementos de N tienen la forma $q2^{k_2}$ ($q \in Q$), por tanto, los elementos de la clase $r+N$ tienen la forma,

$$q2^{k_2} + r, q \in Q. \tag{7}$$

Luego, el problema de recorrer cada elemento de cada clase de equivalencia se reduce a fijar primero un elemento de $\mathbb{Z}_{2^{k_2}}$ y luego recorrer cada elemento del conjunto Q, para buscar una clave consistente en G_K mediante (7). Los elementos del conjunto Q tendrán longitud de bloque $k_d = k_1 - k_2$ y cada clase tendrá 2^{k_d} elementos. Haremos referencia a esta metodología como TBB. El problema en estas metodologías

es la elección de k_2, pues es el parámetro que determina la cantidad de clases de equivalencia y por tanto, la cantidad de elementos dentro de estas, y no hay una forma universal de elegirlo. Si en G_K, k_2 aumenta, las clases tienen menos elementos, pero hay más clases, por el contrario, si disminuye, también lo hacen la cantidad de clases, pero aumentan la cantidad de elementos de cada una. Algo similar ocurre en la primera metodología. Para más detalles ver [4] y [11].

1.2 Problema de la pertenencia de las claves a clases de equivalencia

Sean M un texto plano, K una clave, y C el texto cifrado de M con K (con independencia del cifrado que se use). Sea $\zeta(n) \in G_K$ la clase de equivalencia de $n \in \mathbb{F}_2^{k_1}$ en G_K. Sea $P_{\zeta(n)}(m)$ la probabilidad de que $m \in \mathbb{F}_2^{k_1}$ pertenezca a $\zeta(n)$ (notar que $m \in \zeta(n) \Leftrightarrow \zeta(m) = \zeta(n)$). Entonces, el Problema Probabilístico de Pertenencia (PPP) es: Dado M y C (uno o varios pares), con $C \in \mathbb{F}_2^h$, $h \in \mathbb{Z}_{>0}$, tal que, $|\mathbb{F}_2^h| \leq |\mathbb{F}_2^{k_1}|$. Calcular $P_{\zeta(C)}(K)$.

Teorema (Equivalencia de clases)**.** *Dado $k_1, k_2, h \in \mathbb{Z}_{>0}$, $C \in \mathbb{F}_2^h$ y $K \in \mathbb{F}_2^{k_1}$ tal que, $|\mathbb{F}_2^h| \leq |\mathbb{F}_2^{k_1}|$. Las tres afirmaciones siguientes son equivalentes en G_K:*

a) $\zeta(C) = \zeta(K)$.

b) $C \equiv K \equiv r \ (mod\, 2^{k_2})$.

c) Las últimas k_2 componentes de C y K son iguales.

A partir de este teorema se tiene el siguiente corolario,

Corolario (Probabilidad de pertenencia)**.** *Dado C y k_2, la probabilidad de que K pertenezca a la misma clase de C es igual a $\frac{1}{2^{k_2}}$. O sea,*

$$P_{\zeta(C)}(K) = \frac{1}{2^{k_2}}. \tag{8}$$

La aplicación de este corolario es útil cuando se tiene más de un texto cifrado en la práctica. En este caso, el resultado más interesante, consecuencia de lo antes expuesto, es lo siguiente. Supongamos que se tienen w textos cifrados, entonces,

$$n_w = wP_{\zeta(C)}(K) = \frac{w}{2^{k_2}}, \tag{9}$$

donde n_w es la cantidad teórica de textos cifrados de los w iniciales a cuyas clase pertenece K para un valor k_2 previamente fijado. Esto implica en particular que para un ataque no hace falta buscar en todas las clases de G_K para cada uno de los textos cifrados, sino, que eligiendo un buen valor para k_2, basta con buscar en la misma clase del texto cifrado, ya que de los w, según (9), es probable que en al menos n_w textos la clave se encuentre en dichas n_w clases correspondientes. Para esto es necesario que por lo menos $n_w \geq 1$, lo que implica que $w \geq 2^{k_2}$. Para más detalles ver [13] y [12].

Estos resultados, en particular, sugieren varias formas de elegir k_2 en dependencia de los datos que se tengan, de la capacidad de cálculo, y de lo que se desee hacer. En particular, una vía es que si se obtuvo conocimiento de los últimos l bits de la clave K, entonces se sabe que la clase r a la que pertenece K es el resultado de la conversión a decimal de ese bloque de l bits, y el cálculo de G_K se haría con $k_2 = l$. Otra forma está en dependencia del valor de w, de donde se elegiría k_2 de tal forma que $n_w \geq 1$, y en dependencia de la capacidad de cálculo y tiempo disponible. La idea de este trabajo es estudiar algunos parámetros que permitan ir mejorando los criterios de la elección de los mismos en el momento de realizar pruebas o atacar ciertos cifrados mediante el AG y en particular usando el PPP.

2. Estudio de parámetros

2.1 Propuesta de otras funciones de aptitud

La justificación de por qué la necesidad de buscar otras funciones de aptitud, se precisará en la última sección, en lo referente al ataque al cifrado HTC usando el AG mediante el PPP. No obstante, la idea está en el problema de que muchas veces el AG no encuentra la clave, aun y cuando la busca en la clase donde ella se encuentra. Esto trae consigo que tenga que buscar en otra. En este sentido, sería interesante estudiar otras funciones de aptitud que sean más efectivas en su búsqueda en cada clase.

Sean E un cifrado en bloques con longitud de texto plano y cifrado n, definido como en (6), T un texto plano, K una clave, y C el correspondiente texto cifrado, tal que, $C = E(K,T)$. Sean además Y_d, la correspondiente conversión a decimal del bloque binario Y, y,

$$D : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n,$$

la función de descifrado de E, tal que, $T = D(K,C)$. Entonces, la función de aptitud con la que se ha estado trabajando y que toma como base la distancia de Hamming d_H, para un cierto individuo X de la población es,

$$F_1(X) = \frac{n - d_H(C, E(X,T))}{n},$$

que mide la cercanía entre los textos cifrados C y el texto que se obtiene de cifrar T con la probable clave X (ver [3]). Una función parecida a esta es la que mide la cercanía entre los textos planos,

$$F_2(X) = \frac{n - d_H(T, D(X,C))}{n}.$$

Otra función que sigue la idea de comparar los textos en binario con d_H es la ponderación de F_1 y F_2. Sean $\alpha, \beta \in [0,1] \subset \mathbb{R}$, tal que, $\alpha + \beta = 1$, entonces, esta función quedaría definida de la siguiente manera,

$$F_3(X) = \alpha F_1(X) + \beta F_2(X).$$

Es interesante notar que F_3 consume más tiempo que cada función por separado, pero la idea es tener más efectividad en la búsqueda de la clave.

Las funciones de aptitud que en este punto se proponen, están basadas en medir la cercanía de los textos planos y cifrados pero en decimal. La primera función se define del modo siguiente,

$$F_4(X) = \frac{2^n - 1 - |C_d - E(X,T)_d|}{2^n - 1}.$$

Notar que si los textos cifrados son iguales, $C_d = E(X,T)_d$, entonces, $|C_d - E(X,T)_d| = 0$, lo que implica que $F_4(X) = 1$. O sea, si son iguales, entonces la función de aptitud toma el mayor valor. Por el contrario, la mayor diferencia es lo más alejado que pueden estar, o sea, $C_d = 2^n - 1$, y $E(X,T)_d = 0$, y por tanto, $F_4(X) = 0$.

La siguiente es una ponderación de las funciones F_1 y F_4,

$$F_5(X) = \alpha F_1(X) + \beta F_4(X).$$

Ambas funciones tienen en común que miden la cercanía entre los textos cifrados. Esto no es una ambigüedad, ya que, por ejemplo, si C y $E(X,T)$ se diferencian en dos bits, la función F_1 siempre tendrá el mismo valor sin importar quiénes sean estos dos bits. Por el contrario, no es lo mismo en F_4 si los bits son los dos más o menos significativos, pues los números no son los mismos en su equivalente en decimal.

La siguiente función mide la cercanía en decimal de los textos planos,

$$F_6(X) = \frac{2^n - 1 - |T_d - D(X,C)_d|}{2^n - 1}.$$

Finalmente las funciones F_7, F_8 y F_9 se definen con respecto a las anteriores de la siguiente forma,

$$F_7(X) = \alpha F_2(X) + \beta F_6(X),$$

$$F_8(X) = \alpha F_4(X) + \beta F_6(X),$$

$$F_9(X) = \alpha_1 F_1(X) + \alpha_2 F_2(X) + \alpha_3 F_4(X) + \alpha_4 F_6(X).$$

Donde $\alpha_i \in [0,1] \subset \mathbb{R}, i \in \{1,2,3,4\}$, y, $\sum_{i=1}^{4} \alpha_i = 1$. Con esto se garantiza que en general cada $F_j(X) \in [0,1] \subset \mathbb{R}, j \in \{1,2,3,4,5,6,7,8,9\}$.

Con el objetivo de comparar estas funciones se realizaron experimentos de ataque al cifrado AES(3) para las dos metodologías de partición del espacio de las claves. Se usó una PC con procesador Inter(R) Core(TM) i3-4160 CPU @ 3.60GHz (4 CPUs), y 4GB de RAM. Para los resultados se midió el tiempo medio que tardaron en encontrar la clave, el número medio de generaciones en que se encontró, el porciento de fallos (en una cantidad de ataques realizados), y un parámetro llamado Efectividad, E_{F_i}, que hace una ponderación de los tres criterios anteriores.

Definición (Efectividad de funciones de aptitud)**.** *Sean dados $\mu_1, \mu_2, \mu_3 \in [0,1] \subset \mathbb{R}, \mu_1 + \mu_2 + \mu_3 = 1$, t_{F_i}, $i = \overline{1, \cdots, k}$, el tiempo que demora el AG en encontrar la clave con F_i, en un promedio de g_{F_i} generaciones, y p_{F_i} el porciento de intentos en los que el AG no encontró la clave con F_i. Entonces, la efectividad, E_{F_i}, de la función de aptitud F_i con respecto a las demás $k-1$ funciones $F_j, j \neq i$, se define como,*

$$E_{F_i} = 1 - \left(\mu_1 \frac{t_{F_i}}{\sum_{\gamma=1}^{k} t_{F_\gamma}} + \mu_2 \frac{g_{F_i}}{\sum_{\gamma=1}^{k} g_{F_\gamma}} + \mu_3 \frac{p_{F_i}}{\sum_{\gamma=1}^{k} p_{F_\gamma}}\right)$$

Una generalización de esta definición para más parámetros se realiza de forma equivalente. En la tabla 1 se presentan los resultados de la comparación de las diferentes funciones de aptitud para la metodología BBM de partición del espacio, en este caso, $k = 9$. Se tomó $\alpha = \beta = 0{,}5$, y cada $\alpha_i = 0{,}25$. Para calcular E_{F_i} se tomaron los valores $\mu_1 = 0{,}33$, $\mu_2 = 0{,}33$ y $\mu_3 = 0{,}34$ para t_{F_i}, g_{F_i} y p_{F_i} respectivamente. Ordenando las

Tabla 1. Comparación de las funciones de aptitud, con BBM

F_i	Tiempo	Generaciones	Fallos (%)	E_{F_i}
F_1	5.233	121.2	60	0.8731
F_2	5.402	108.4	50	0.8870
F_3	11.101	117.4	50	0.8584
F_4	4.764	109.2	40	0.8995
F_5	9.451	109.8	30	0.8885
F_6	3.126	63.4	20	0.9433
F_7	12.424	121.3	50	0.8511
F_8	7.054	77.1	10	0.9309
F_9	15.811	87.7	30	0.8682

F_i con respecto a la efectividad, las primeras 5 serían F_6, F_8, F_4, F_5 y F_2. Es de notar que de las tres primeras que usan solo la distancia de Hamming, solo aparece F_2.

En el caso de la comparación de dichas funciones para la metodología TBB de partición del espacio de las claves y búsqueda en G_K, los resultados de los experimentos se presentan en la tabla 2. En este caso, ordenando las funciones

Tabla 2. Comparación de las funciones de aptitud, con TBB

F_i	Tiempo	Generaciones	Fallos (%)	E_{F_i}
F_1	3.688	83.1	20	0.9278
F_2	5.353	109.1	60	0.8633
F_3	11.403	122.9	40	0.8536
F_4	3.226	67.8	30	0.9240
F_5	7.147	83.4	10	0.9235
F_6	4.871	96.2	40	0.8939
F_7	10.694	113.1	20	0.8840
F_8	8.354	92	20	0.9029
F_9	16.876	95.7	50	0.8270

por su efectividad, las 5 primeras serían F_1, F_4, F_5, F_8 y F_6. Nuevamente aparece una sola función de las tres primeras, en este caso, F_1, y las otras repiten.

2.2 Ataque Directo y por Eliminación de Clases

En esta sección se trabajará con la metodología TBB. Nos centramos en una vía para la elección de k_2, el Ataque Directo, y terminamos con una metodología de ataque por eliminación de clases que permite restringir, con cierta probabilidad, la cantidad de clases a recorrer. Los experimentos de esta sección se realizan con el cifrado HTC, una Red de Sustitución Permutación de 16 bits de longitud de bloque, para más detalles ver [5] y [9].

Supongamos que se tienen w pares de textos planos y sus correspondientes textos cifrados, y es la única información que se tiene para buscar la clave, de la cual se desconoce todo. Lo primero es elegir k_2, que como el ataque es en G_K, entonces representa la cantidad de clases en las que se dividirá el grupo cociente. Sabemos, según el PPP, que hay un rango para seleccionar k_2, cumpliendo con que $n_w > 1$, según (9) y en dependencia de la capacidad de cálculo y tiempo. Luego de elegir k_2 y calcular G_K se debe elegir una clase para buscar la clave, tema resuelto según PPP, con lo cual, se buscará la clave en la misma clase a la que pertenece el texto cifrado correspondiente. Luego, solo quedaría ir recorriendo cada uno de los pares texto plano y texto cifrado y buscar la clave en la misma clase del texto cifrado.

Para los experimentos se usó una PC con procesador Inter(R) Core(TM) i5-3340 CPU @ 3.10GHz (4 CPUs) y 4GB de RAM. Partimos de $w = 100$ pares de textos, por tanto, se puede elegir un valor de k_2 entre 1 y 6 (de los 16 posibles), ya que si $k_2 = 7$, entonces, $n_w = 100/2^7 = 100/128 < 1$, con lo cual sería poco probable encontrar un texto cifrado en cuya clase esté la clave. Pero de los valores de 1 a 6, este último está en una situación parecida, ya que $n_w = 100/2^6 = 100/64 \approx 1{,}6$. En cambio, parece haber mejores resultados para los valores de 1 a 5. Esto se comprobó para los valores de k_2 de 2 a 10, realizando el ataque buscando por cada uno de los w pares hasta encontrar la clave, lo que hemos llamado Ataque Directo. Para cada uno de los intentos en cada valor de k_2, y para valores diferente de este, los w pares de textos son diferentes cada vez. Los resultados se pueden ver en la tabla 3. Es

Tabla 3. Ataque Directo al HTC con $w = 100$

k_2	No. de clases	Tiempo	Generac.	Fallos (%)
2	6.3	304.975	68.1	0
3	11.2	257.25	45.5	0
4	19	130.158	9.1	0
5	62.8	217.516	12.7	30
6	67.5	111.559	8.3	60
7	56.9	44.559	3.1	30
8	86.9	28.586	2.2	60
9	86.7	17.253	1.7	70
10	101	10.116	2	100

importante señalar que la función de aptitud que se usó fue F_1. En la tabla se puede apreciar cómo disminuye el tiempo a medida que aumenta k_2, lo cual es claro debido a que con el aumento de k_2 también aumentan la cantidad de clases, y por tanto, disminuye la cantidad de elementos dentro de las mismas, lo que hace que el AG termine más rápido. Es interesante notar que los fallos comienzan a aparecer a partir de $k_2 = 5$, y van gradualmente en aumento a medida que lo hace k_2. O sea, aunque hay un rango para elegir de 1 a 16, los mejores valores están hasta 4 y máximo 5, con respecto a los 100 textos cifrados que se están usando. Este punto corrobora la idea de solución al problema de la elección de k_2.

Una desventaja que presenta hacer el ataque de esta manera, es que se recorren, por lo general, más clases de las que tiene la partición. Por ejemplo, con todo y que elijamos $k_2 = 4$, es de esperarse como promedio que recorra 19 clases (pares de textos planos y cifrados) hasta encontrar la clave, según la tabla 3. Pero G_K solo tendría $2^{k_2} = 2^4 = 16$ clases, y se están recorriendo más. Lo que sucede es, por una parte, que hay varios textos cifrados que pertenecen a una misma clase donde no está la clave, por lo que el AG recorrerá esa clase varias veces; y por otro lado, que el AG, con la función de aptitud que se está usando, F_1, muchas veces no es capaz de encontrar la clave buscando en la clase correcta, por eso sigue buscando el siguiente texto cifrado hasta encontrarla, y se repiten algunas clases.

Es aquí donde surge la necesidad de buscar otras funciones de aptitud que hagan que el AG sea más efectivo y disminuya la cantidad de clases. Aunque es algo que no basta, sino, que es necesario ver si la forma de recorrer los textos cifrados se puede mejorar. Esto se verá más adelante, en el Ataque por Eliminación de Clases.

2.2.1 Ataque por Eliminación de Clases

Supongamos dados w textos cifrados y k_2, de tal forma que $n_w \geq 1$, o sea, $w \geq 2^{k_2}$. Se tienen 2^{k_2} clases de equivalencia, y los w textos se pueden agrupar en dichas clases. Para saber la probabilidad de que de los w, w_1 pertenezcan a la clase X_1, w_2 a la X_2, $\cdots$, $w_{2^{k_2}}$ a la $X_{2^{k_2}}$, se puede utilizar la Distribución Multinomial,

$$P(X_1 = w_1, \cdots, X_{k_2} = w_{k_2}) = \frac{w!}{w_1!w_2!\cdots w_{2^{k_2}}} p_1^{w_1} \cdots p_{k_2}^{w_{k_2}}, \tag{10}$$

donde $\sum_{i=1}^{2^{k_2}} w_i = w$.

Todas las clases tienen igual probabilidad $p = \frac{1}{2^{k_2}}$ de ser elegidas, por tanto, (10) se reduce a,

$$P(X_1 = w_1, \cdots, X_{k_2} = w_{k_2}) = \frac{w!}{w_1!w_2!\cdots w_{2^{k_2}}} p^w. \tag{11}$$

La esperanza matemática de X es, $E(X) = wp = w\frac{1}{2^{k_2}}$, que equivale a n_w. Esto implica que lo más probable al organizar los w textos cifrados en las 2^{k_2} clases, es que se espere que las clases tengan en promedio n_w elementos y alrededor de este número esté la mayor concentración. La probabilidad de que hayan muchos más elementos en cada clase, o muy pocos, va disminuyendo a medida que se aleja de la media n_w. De ser así, la clave debe aparecer más frecuentemente si se busca en las clases que tienen una cantidad de elementos cercana a n_w.

Para comprobar esto, se tomó $k_2 = 4$, y se generaron 19200 textos cifrados, agrupados en 200 pruebas de 96 textos cada una. La idea es, en cada prueba, agrupar los 96 textos en las $2^4 = 16$ clases y verificar cuántos elementos tiene la clase a la que pertenece la clave. Se usó una PC Laptop con procesador: Intel(R) Celeron(R) CPU N3050 @1.60GHz (2 CPUs), ∼1.6GHz y 4GB de RAM. En estas 200 pruebas, los resultados se muestran en la tabla 4. Las primera y tercera

Tabla 4. Veces que aparece la clave alrededor de n_w

No. Elem	No. Aparic	No. Elem	No. Aparic
0	2	9	16
1	3	10	5
2	8	11	4
3	21	12	5
4	25	13	0
5	32	14	0
6	25	15	0
7	31	16+	0
8	23	–	–

columnas indican la cantidad de elementos que tiene la clase, y las segunda y cuarta, la cantidad de veces que se encontraba la clave. Como se puede apreciar, la mayor cantidad de claves está concentrada alrededor del 6 (ya que $n_w = 96/16 = 6$). Por ejemplo, en el intervalo de las clases que tienen de 3 a 9 elementos, se encuentran 173 claves, el 86.5 % de las 200 totales.

En este sentido, la idea del Ataque por Eliminación de Clases es agrupar los w textos cifrados en clases de equivalencia, luego elegir un intervalo,

$$[a,b] \subset \mathbb{N},$$

donde a y b representan la cantidad de textos cifrados en las clases, luego de ser agrupados, en un entorno de n_w, con $a \leq n_w \leq b$. De tal forma que la longitud del intervalo sea menor que la cantidad de clases al menos una vez,

$$\eta(b-a+1) < 2^{k_2}, \eta \in \mathbb{N} \tag{12}$$

Con (12) se garantiza una *condición de eliminación*, ya que se está dando la posibilidad de recorrer η veces las $b-a+1$ clases contenidas en el intervalo, sin aun llegar a las 2^{k_2} totales como caso ideal, y se parte de esa base aunque no siempre ocurrirá así. Las etapas del Ataque por Eliminación de Clases (AEC) se podrían resumir en,

I: Entrada: w pares de texto plano y cifrados.

II.1: Elegir k_2 y calcular n_w.

II.2: Agrupar los w pares en las 2^{k_2} clases de equivalencia.

II.3: Elejir η y $[a,b]$ satisfasciendo (12).

III: Buscar la clave con el AG en las clases que en la agrupación tienen una cantidad de textos cifrados dentro de $[a,b]$, repetir η veces si no se encuentra.

Se realizaron experimentos aplicando el AEC al HTC en la misma PC Laptop para los resultados de la tabla 4 y los datos que se han estado tratando: $w = 100$, $k_2 = 4$, $2^{k_2} = 16$ clases, $\eta = 2$ y se eligió el intervalo $[3,9]$, de tal forma que, $2*(9-3+1) = 14 < 2^4 = 16$. Se probó con dos funciones de aptitud, F_1, la que se ha estado utilizando y F_5. Para cada una de las mismas se hicieron 40 corridas, lo que da un total de 8000 pares de textos planos y cifrados. Los resultados se muestran en la tabla 5. Notar cómo se reduce en promedio la

Tabla 5. Ataque por Eliminación de Clases al HTC

F_i	No. de clases	Tiempo	Generac.	Fallos (%)
F_1	8.83	114.26	18.999	40
F_5	10.4	193.59	12.44	15

cantidad de clases necesarias para encontrar la clave. Con F_1 prácticamente solo hizo falta recorrer la mitad de las clases como promedio, 8.83 de 16, algo más necesitó F_5 aunque con resultados bastante parecidos, 10.4 de 16. F_1 tiene un porciento de aciertos del 60 %, sin embargo, la función F_5, es más efectiva, con un 85 % de aciertos.

3. Conclusiones

En el presente artículo se estudiaron varios aspectos sobre algunos parámetros del Algoritmo Genético y el ataque a cifrados en bloque. Se propusieron varias funciones de aptitud con buenos resultados en los experimentos con respecto a las funciones que se estaban usando. Se propusieron dos metodologías de ataque que unen el AG con el PPP: el Ataque Directo y el Ataque por Eliminación de Clases, este último con la posibilidad de obtener resultados aceptables sin recorrer todas las clases necesarias.

Para trabajos futuros es interesante seguir avanzando en el estudio de ciertos parámetros y criterios de elección que permitan mejorar la efectividad de los ataques usando el AG y el PPP.

Referencias

[1] Adwan, Al, M. Al Shraideh y M.R.S. Al Saidat: *A Genetic Algorithm Approach for Breaking of Simplified Data Encryption Standard*. International Journal of Security and Its Applications, 9(9):295–304, 2015. http://www.sersc.org/journals/IJSIA/vol9no92015/26.pdf.

[2] Baragada, SR. y P.S. Reddy: *A Survey of Cryptanalytic Works Based on Genetic Algorithms*. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS). ISSN 2278-6856, 2(5), September-October 2013. http://www.ijettcs.org/Volume2Issue5/IJETTCS-2013-08...

[3] Borges-Trenard, M.A., M. Borges-Quintana, A. Donatien-Charón y L. Monier-Columbié: *Nueva función de aptitud en el criptoanálisis genético a cifrados en*

bloques. Congreso Internacional COMPUMAT. La Habana, Cuba, 2017.

[4] Borges-Trenard, M.A., M. Borges-Quintana y L. Monier-Columbié: *An application of genetic algorithm to cryptanalysis of block ciphers by partitioning the key space*. Journal of Discrete Mathematical Sciences & Cryptography, 2019. DOI: 10.1080/09720529.2019.1649028.

[5] Borges-Trenard, M.A. y L. Monier-Columbié: *AES(t): Una versión parametrizada del AES*. Congreso Internacional COMPUMAT. La Habana, Cuba, 2015.

[6] Brown, J.A., S.K. Houghten y B. Ombuki-Berman: *Genetic Algorithm Cryptanalysis of a Substitution Permutation Network*. IEEE Symposium on Computational Intelligence in Cyber Security, páginas 115–121, 2009.

[7] Delman, Bethany: *Genetic algorithms in cryptography*. Thesis. Rochester Institute of Technology, RIT Scholar Works, 2004. http://scholarworks.rit.edu/cgi/viewcontent.cgi?article=6...

[8] Garg, P., S. Varshney y M. Bhardwaj: *Cryptanalysis of Simplified Data Encryption Standard Using Genetic Algorithm*. American Journal of Networks and Communications, 4(3):32–36, 2015. http://article.sciencepublishinggroup.com/pdf/10.11648.j...

[9] Howard, Heys M.: *A tutorial on Linear and Differential Cryptanalysis*. Cryptologia, 26(3):189–221, 2002.

[10] Khan, A.H., A.H. Lone y F.A. Badroo: *The Applicability of Genetic Algorithm in Cryptanalysis: A Survey*. International Journal of Computer Applications, 130(9), 2015. http://www.ijcaonline.org/research/volume130/number9/...

[11] Tito, Osmani, Miguel A. Borges-Trenard y Mijail Borges-Quintana: *Ataques a cifrados en bloques mediante búsquedas en grupos cocientes de las claves*. Revista Ciencias Matemáticas, 33(1), 2019.

[12] Tito, Osmani, Miguel A. Borges-Trenard y Mijail Borges-Quintana: *Sobre la partición del espacio de las claves*. Ciencia e Innovación Tecnológica. Editorial Académica Universitaria & Opuntia Brava, VIII:401–409, 2019.

[13] Tito, Osmani, Miguel A. Borges-Trenard y Mijail Borges-Quintana: *Sobre la pertenencia de las claves a clases de equivalencia en G_K*. XVI Congreso Internacional COMPUMAT, 2019.

[14] Vimalathithan, R. y M.L. Valarmathi: *Cryptanalysis of DES using Computational Intelligence*. European Journal of Scientific Research, ISSN 1450-216X, 55(2):237–244, 2011.