Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

Tipo de artículo: Artículos originales
Temática: Criptoanálisis a cifrados en bloques
Recibido: 06/11/2022 | Aceptado: 18/12/2022 | Publicado: 29/01/2023

Mejorando la búsqueda de soluciones de sistemas MRHS mediante el Algoritmo Genético

Improving search of solutions of MRHS systems using the Genetic Algorithm

Osmani Tito-Corrioso 0000-0003-4231-7674[1*]
Mijail Borges-Quintana 0000-0002-5656-0037[2]
Miguel A. Borges-Trenard 0000-0001-7154-7730[3]

[1]Departamento de Matemática - Física Aplicada, Facultad de Ingeniería Industrial, Universidad de Matanzas, Autopista a Varadero km 3.5, Matanzas, Cuba. osmani.tito@umcc.cu
[2]Departamento de Matemática, Facultad de Ciencias Naturales y Exactas, Universidad de Oriente, Av. Patricio Lumumba s/n, Santiago de Cuba 90500, Cuba. mijail@uo.edu.cu
[3]Doctorate in Mathematics Education, Universidad Antonio Nariño, Bogotá 111321, Colombia. borgestrenard2014@gmail.com

[*]Autor para correspondencia: (osmani.tito@umcc.cu)

RESUMEN

El Ataque Algebraico es uno de los métodos más usados en el criptoanálisis a cifrados en bloque, enfocado principalmente en transformar el cifrado en un sistema de ecuaciones. Esto se puede realizar de diferentes formas, en particular, el foco de este trabajo son los sistemas de ecuaciones del tipo MRHS, Multiple Right Hand Side, una forma especial para transformar los cifrados en bloque. Los sistemas MRHS constituyen un método alternativo para el trabajo algebraico con estos cifrados. La particularidad principal de estos sistemas es que el término independiente no es único para cada ecuación, sino, que es un conjunto de términos independientes. En ese sentido, en este trabajo se presenta un primer estudio sobre el tema, y se propone el uso del Algoritmo Genético, AG, como método de solución de estos sistemas de ecuaciones. Por otro lado, se proponen tres funciones de aptitud que permiten conectar los sistemas MRHS con el AG. Con los experimentos se

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

obtuvieron buenos resultados en la solución de los sistemas MRHS mediante el AG, mostrando, además, que el AG no solo los soluciona, sino, que muchas veces encuentra varias soluciones.

Palabras clave: Criptoanálisis Algebraico; Cifrados en Bloque; Funciones de Aptitud; Sistemas MRHS; Algoritmos Genéticos.

ABSTRACT

The Algebraic Attack is one of the more used methods in the cryptanalysis of block ciphers, focused principally in to transform the cipher into a system of equations. This can be accomplished of different ways, in particular, the focus of this work are the systems of equations of the type MRHS, Multiple Right Hand Side, an especial form to transform the block ciphers. MRHS systems constitute an alternative method for the algebraic work with this ciphers. The main particularity of these systems is that the independent term is not unique for each equation, but, that it is a set of independent terms. In that direction, in this work we present a first study about this topic, and we propose the use of the Genetic Algorithm, GA, as a method of solution of these systems of equations. In addition, we propose three fitness functions that allow connecting systems MRHS with the GA. With the experiments we obtain good results in the solution of the MRHS systems with the GA, showing, besides, that the GA did not only solves MRHS systems, but, that many times it find several solutions.

Keywords: Algebraic Cryptanalysis; Block Ciphers; Fitness Functions; MRHS Systems; Genetic Algorithms.

Introduction

The Genetic Algorithm (GA) is a heuristic optimization method based on biologic genetic evolution. A successful application in cryptanalysis to block ciphers has been achieved. Some of the research carried out in this direction is mentioned below. The aim of (Mittal and Gupta, 2019) is to establish an algorithm for encryption and decryption of a message based on symmetric key cryptosystem involving GA and its operators. In (Abduljabbar et al., 2021) a fast technique for text encryption depending on GA and its operators is presented. In (Nofal, 2022) an optimization problem, associate with the Internet of Things, is solved using GA. In (Aas-

hiqbanu et al., 2022) authors propose a hybrid Telugu cryptography based on GA by mapping it in Unicode. Confusion and diffusion are performed to strengthen the algorithm's security in four stages: pre-processing, reshape, crossover and mutation. The papers (Qiu and Wang, 2022) and (Artuğer and Özkaynak, 2022) are similar: authors propose to use the GA to search cryptographically significant substitution box: S-boxes. The idea is to find resistant S-boxes against different attacks. Other details about GA can be seen in (Katoch et al., 2021), (Tito-Corrioso et al., 2021) and (Cao et al., 2022).

On the other hand, we focus on algebraic cryptanalysis, characterized by the formulation of the attack problem as the solution of a system of equations. Representing ciphers as a system of equations can be done in different ways, in particular, the focus of this work is on Multiple Right Hand Side, MRHS, equations systems, introduced in (Raddum and Semaev, 2006) These systems are an alternative method for algebraic work with block ciphers. Its main peculiarity is that the independent term is not unique for each equation, but rather, it is a set of independent terms. Actually there are several paper about this topic. For example, in (Zajac and Špaček, 2019) authors propose a new concept of (post-quantum) digital signature algorithm derived from a symmetric cipher which security is based on the difficulty of solving MRHS equations and key derivation is based on this systems. The focus in (Indrøy and Raddum, 2021) is the problem of evaluating a block cipher's strength against differential or linear cryptanalysis via Compressed Right-Hand Side equations, which are MRHS equations where the right-hand sides are stored using a binary decision diagram instead of multiple independent vectors. Further details on the theory of MRHS systems can be found in (Indrøy, 2018), (Matheis et al., 2019) and (Zajac, 2021).

This paper presents a first approach to the subject of MRHS systems and the use of the Genetic Algorithm as a method of solving them. Searching for the solution in the entire space is increasingly difficult as the number of variables increases, for this reason, the use of two partition methodologies is also proposed of the space, which allow to focus the search on one or several subsets of the complete space. On the other hand, three fitness functions are proposed that allow connecting the MRHS systems with the GA. The experiments showed good results in the solution of the MRHS systems by GA.

Materials and methods

Genetic Algorithm

GA is a heuristic optimization method. We assume that the reader knows the general ideas of how the Genetic Algorithm works. This section we will briefly describe the GA scheme used in this work.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

Algorithm 1 Genetic Algorithm

Input: m, number of individuals in the population; F, fitness function; g, number of generations; s, number of individuals selected to mate.

Output: the individual with the highest fitness function as the best solution.

1: **Generate** randomly an initial population P_i with m individuals, possible solutions.
2: **Calculate** the fitness of each individual of P_i with F.
3: **while** no solution found or g generations not reached **do**
4: **Select** s parents of P_i.
5: Perform the **Crossover** of the s selected parents and generate offspring pairs.
6: **Mutate** each of the resulting descendants.
7: **Compute** the fitness of each of the offspring and their mutations with F.
8: Using the Tournament Method between two, based on the aptitudes of the parents and offspring, decide what will be the new population P_{i+1} for the next generation, selecting two individuals at random each time and choosing the higher fitness.
9: **end while**

The individuals of the populations will be elements of the key space taken as binary blocks. By **Selecting** the s parents, a subset S of P_i is obtained. These parents are selected by the Tournament Method between two, selecting two individuals at random and choosing the one with the highest aptitude. Elements of S are crossed, and descendants are added to P_i if they are not members. For **Crossover** the two-point crossover will be used, and the probability of two individuals crossing over was set to 0.6 for all experiments. The **Mutate** operation consists of making changes to at most three random components of the binary block, with a mutation ratio set to 0.2 in all experiments. An individual x is better adapted than another y, if it has greater fitness, that is, if $F(x) > F(y)$. The fitness functions with which we make the connection between the GA and the MRHS equations systems will be defined in next sections. For the GA specification to block ciphers, see Section 3 of (Borges-Trenard et al., 2019).

This application of GA in Cryptography constitutes a known-plaintext attack model proposal, in which the attacker knows a set of plaintexts with their corresponding encrypted texts. The attack's goal is to find the key with which the plaintexts were encrypted.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

Key space partition methodologies

Key space partition methodologies allow the GA to work on a certain subset of the set of admissible solutions as if it were the full set. The importance of this fact is that it reduces the size of the search space and gives the heuristic method a higher chance of success, assuming that the fittest individuals are found in the selected subset. On the other hand, the partition into equivalence classes allows it to use the GA or another algorithm in parallel, in several classes simultaneously.

The first methodology is the BBM, proposed in (Borges-Trenard et al., 2019). Let $\mathbb{F}_2^{k_1}$ be the space of keys of length $k_1 \in \mathbb{Z}_{>0}$. It is known that $\mathbb{F}_2^{k_1}$ has cardinal 2^{k_1} and therefore there is a one-to-one correspondence between $\mathbb{F}_2^{k_1}$ and the interval $[0, 2^{k_1}-1] \subset \mathbb{Z}_+$. If an integer k_2 is set, $(1 \leq k_2 < k_1)$, then the key space can be represented by the numbers, $q2^{k_2}+r$, where $q \in [0, 2^{k_1-k_2}-1] \subset \mathbb{Z}_+$, and, $r \in [0, 2^{k_2}-1] \subset \mathbb{Z}_+$. In this way the key space is divided into $2^{k_1-k_2}$ blocks, determined by the quotient in the division algorithm dividing by 2^{k_2}, and, within each block, the corresponding key is determined by its position, which is given by the remainder r. The main idea is to position yourself in a block, given by q, and move within that block through the elements, given by r, using the GA.

The next methodology is the TBB, proposed in (Tito-Corrioso et al., 2019) and (Tito-Corrioso et al., 2021). This methodology is based on the definition and calculation of the quotient group of keys G_K whose objective is to partition $\mathbb{F}_2^{k_1}$ into equivalence classes. It is known that $\mathbb{F}_2^{k_1}$, as an additive group, is isomorphic to $\mathbb{Z}_{2^{k_1}}$. Let h be the homomorphism defined as follows: $h : \mathbb{Z}_{2^{k_1}} \longrightarrow \mathbb{Z}_{2^{k_2}}$, s.t. $a \longrightarrow a \pmod{2^{k_2}}$, where $k_2 \in \mathbb{Z}_{>0}$ and $0 < k_2 < k_1$. Let us denote by N the kernel of h, then, G_K is the quotient group of $\mathbb{Z}_{2^{k_1}}$ times N, that is, $G_K = \mathbb{Z}_{2^{k_1}}/N$. And finally, $G_K = \{N, 1+N, 2+N, \ldots, (2^{k_2}-2)+N, (2^{k_2}-1)+N\}$.

In this way $\mathbb{Z}_{2^{k_1}}$ is divided into a partition of 2^{k_2} classes given by N. To iterate through the elements of each class, note that $\mathbb{Z}_{2^{k_2}}$ is isomorphic with G_K and the isomorphism corresponds to each $r \in \mathbb{Z}_{2^{k_2}}$ its equivalence class $r+N$ in G_K. Therefore, selecting a class is setting an element $r \in \mathbb{Z}_{2^{k_2}}$. On the other hand, the elements of N have the form $q2^{k_2}$, $q \in \{0, 1, 2, \ldots, 2^{k_1-k_2}-1\}$, therefore, the elements of the class $r+N$ have the form, $q2^{k_2}+r$.

The idea of partitioning in the context of MRHS equations systems is, on the one hand, to decrease the search space for solutions, which in the general case is $\mathbb{F}_2^n$, where n is the number of variables; and on the other hand, allow parallel search in different classes. In this case we speak of space of solutions and not space of keys, although these spaces have the same structure.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

MRHS Equations Systems

Vectors over $\mathbb{F}_2$ will be row vectors. For more details on the theory of MRHS systems see (Raddum and Semaev, 2008) and (Raddum and Zajac, 2018).

DEFINITION 1 *An MRHS equation over $\mathbb{F}_2$ is an expression of the form, $x\mathbf{M} \in S$, where $\mathbf{M}$ is a matrix $n \times l$, and $S \subset \mathbb{F}_2^l$ is a set of vectors of l bits. A vector $x \in \mathbb{F}_2^n$ is said to be a solution of the MRHS equation, if and only if, $x\mathbf{M} \in S$.*

A system of MRHS equations, $\mathscr{M}$, is a set of m MRHS equations with the same dimension n: number of variables. That is, $\mathscr{M} = \{x\mathbf{M}_i \in S_i | 1 \leq i \leq m\}$, where each $\mathbf{M}_i$ is a matrix $n \times l_i$, and $S_i \subset \mathbb{F}_2^{l_i}$. The vector $x \in \mathbb{F}_2^n$ is a solution of the system $\mathscr{M}$, if it is a solution of all the equations in $\mathscr{M}$, this is, $x\mathbf{M}_i \in S_i$, $\forall i \in \{1,2,\ldots,m\}$.

Given a system of equations $\mathscr{M} = \{x\mathbf{M}_i \in S_i | 1 \leq i \leq m\}$, they can be concatenated by the columns all the $\mathbf{M}_i$ matrices, taking into account that they have the same number of rows, variables. The resulting matrix is called the *united system matrix*, and is represented as $\mathbf{M} = [\mathbf{M}_1|\mathbf{M}_2|\ldots|\mathbf{M}_m]$. The columns of $\mathbf{M}$ that correspond to $\mathbf{M}_i$ are called *block*. In the same way, the Cartesian product of all the right sides S_i is carried out, to obtain a single set $S = S_1 \times S_2 \times \cdots \times S_m$. The problem of solving a system of MRHS equations can now be stated as, find some $x \in \mathbb{F}_2^n$, such that, $x\mathbf{M} \in S$.

The ciphers that can be represented as a sequence of linear, or affine, transformations, and substitution layers performed by S-boxes, can be represented by a system of MRHS equations as follows. Let s be the number of S-boxes per round; l_a and l_o the number of input and output bits of the S-boxes; nR the number of rounds; n_B the size of the plaintext and ciphertext; and n_K the key length. The parameters of the MRHS system are: $l_i = l_a + l_o$, $|S_i| = 2^{l_a}$, $m = s \cdot n_R$, and $n = 2n_B + n_K + ml_o$. The vector of variables $x \in \mathbb{F}_2^n$ of the system consists of the n_B bits of the plaintext $(x_1,\ldots,x_{n_B})$, n_K bits of key $(x_{n_B+1},\ldots,x_{n_B+n_K})$, all outputs of the S-boxes $(x_{n_B+n_K+1},\ldots,x_{n-n_B})$, and the n_B bits of the ciphertext $(x_{n-n_B+1},\ldots,x_n)$. If S-boxes are used in the key generation scheme, they are considered as part of the encryption. All input bits of all S-boxes can be described as linear, or affine if there is addition of constants, combinations of the variables defined above. From each S-box i an MRHS equation is created as follows,

$$x\mathbf{M}_i \in \{(0 \oplus c||S(0 \oplus c)), (1 \oplus c||S(1 \oplus c)), \ldots, ((2^{l_a}-1) \oplus c||S((2^{l_a}-1) \oplus c))\} = S_i, \quad (1)$$

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

where c is the constant input part of the affine combination, using the natural conversion between the integers and the vectors of $\mathbb{F}_2$. The first l_a columns of $\mathbf{M}_i$ contain the coefficients of the linear combination of the input to the S-box. The last l_o columns contain a single bit 1 each: $m_{j,t} = 1$ if x_j is the variable for output bit t after multiplication, and $m_{j,t} = 0$ otherwise.

When the objective of building the united matrix of the system is to do an algebraic cryptanalysis, focused mainly on the search for the key, then we assume that the pair of plaintext and ciphertext is known. Thus the system can be reduced by setting the first and last n_B variables to x as follows. If the original joined system is $x\mathbf{M} \in S$, then it can be written as,

$$(p, x', c) \cdot \begin{bmatrix} \mathbf{M}_p \\ \mathbf{M}' \\ \mathbf{M}_c \end{bmatrix} \in S_1 \times S_2 \times \cdots \times S_m, \tag{2}$$

where p and c are the known values of the plaintext and ciphertext pair, and x' are the remaining variables. Calculated, $p\mathbf{M}_p + c\mathbf{M}_c = w = (w_1, w_2, \ldots, w_m)$, the reduced MRHS system is obtained:

$$x'\mathbf{M}' \in (w_1 + S_1) \times (w_2 + S_2) \times \cdots \times (w_m + S_m), \tag{3}$$

where, $w_i + S_i = \{w + v | v \in S_i\}$. Note that the scalar operations are performed on the finite field $\mathbb{F}_2$.

Results and discussion

Solution of MRHS equations systems using GA

To work with MRHS systems, a toolbox of functions were implemented in Maple 17 that serve as a tool for manipulating these systems of equations. Below we describe some of the functions. The GA operations are carried out on the elements of the solution space of the MRHS system without any operation of the MRHS system intervening. Where the MRHS system only comes into play is when the GA assesses the suitability of each item. In this sense, two fitness functions are proposed through which the connection between the GA and the system is made. Given a vector $x \in \mathbb{F}_2^n$, $\mathbf{M}$ and S, the first fitness function is defined as follows:

$$H_1(x) = \frac{l - h_{\min}(x\mathbf{M}, S)}{l}, \tag{4}$$

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

where l is the number of columns in $\mathbf{M}$, and $h_{\min}$ is the minimum Hamming distance, d, between $x\mathbf{M}$ and each of the vectors of S, more formally, $h_{\min}(x\mathbf{M}, S) = \min\{d(x\mathbf{M}, v) | v \in S\}$. Let $[Y]_{dec}$ be the corresponding conversion to decimal of the binary block Y, the second fitness function is:

$$H_2(x) = \frac{2^l - 1 - g(x\mathbf{M}, S)}{2^l - 1}, \tag{5}$$

where, $g(x\mathbf{M}, S) = \min\{\text{abs}([x\mathbf{M}]_{dec} - [v]_{dec}) | v \in S\}$. These functions, in short, what they do is take the smallest distance between $x\mathbf{M}$ and the set S as the best solution. Note that, $H_1(x), H_2(x) \in [0, 1] \subset \mathbb{R}$.

Experiments and results

Experiments were performed on a Laptop PC with processor: Intel(R) Celeron(R) CPU N3050 @1.60GHz (2 CPUs), $\sim$ 1.6GHz and 4GB RAM. The experiment consists of generating 20 MRHS systems randomly, and verifying if the GA is able to find one or several solutions for these systems. This GA search was performed with each of the two fitness functions for all generated MRHS systems. These systems have $n = 16$ variables and 3 MRHS equations each. The 3 blocks of $\mathbf{M}$ have 3, 4 and 5 columns respectively, so $\mathbf{M}$ has a total of 12 columns and $16 \cdot 12 = 192$ items. The sets S_1, S_2 and S_3 have 4, 5 and 6 elements each, so the product set $S = S_1 \times S_2 \times S_3$, has 120 elements. Regarding the GA parameters, the number of individuals in the population is 50. The search will be done by dividing the space with the BBM methodology, where $k_1 = n = 16$, and $k_2 = 12$. With these data, in total, the GA must go through at most 81 generations for each system.

The results obtained for each of the fitness functions are as follows: with the function H_1, an average of 4.1 different solutions per system was found, in an average time of 0.64215 seconds and only reaching 1.05 generations; the H_2 function found an average of 3.4 different solutions per system, in an average time of 3.78125 seconds and in 1.15 generations on average. Neither function was flawed, in the sense that they found at least one solution on each system. As can be seen, the GA not only solves MRHS systems, but is also capable of finding more than one solution.

Note how in the small dimensions of the developed experiments it happens that several generations of the GA are practically not necessary to obtain the solutions. In addition, short times are obtained, which allows evaluating the possibilities of this tool for the cryptanalysis of block ciphers in two fundamental directions. The first is that GA can be an effective method of finding solutions for MRHS systems. The second is that the

computation time behaves well according to the algebraic nature of the problem under study.

Improving search in S

It has already been shown that with the functions H_1 and H_2 solutions of the MRHS systems are found. However, they present difficulties as the number of elements in S increases. The problem is that, in order to evaluate them, it is necessary to go through S completely. As $|S|$ becomes larger, the time consumed by each function, and therefore by the GA, also grows considerably. In this sense, in this section a methodology is proposed to improve the evaluation and search of the elements of S.

It proceeds as follows. A set S' is calculated whose elements s_i, $i = \overline{1,|S'|}$, are those of S converted into their decimal expression. Then S' is ordered such that, if $i \leq j$, then, $s_i \leq s_j$, $\forall s_i, s_j \in S'$. The improved Binary Search Algorithm will be applied to S', to search for $[x\mathbf{M}]_{dec}$. The objective is to obtain n such that, $s_n < [x\mathbf{M}]_{dec} \leq s_{n+1}$. That is, get the smallest range of elements where $[x\mathbf{M}]_{dec}$ is found, which includes the case where the element s_{n+1} is same as searched. Then, the fitness function $H_2(x)$ is applied, passing the set $L = \{s_n, s_{n+1}\}$ to the distance g, or be, $g(x,L)$. To avoid confusion in the notation and in the procedure, the function will be denoted as H_3:

$$H_3(x) = \frac{2^l - 1 - g(x\mathbf{M}, L)}{2^l - 1}, \tag{6}$$

where, $g(x\mathbf{M}, L) = \min\{\text{abs}([x\mathbf{M}]_{dec} - v) | v \in L\}$.

Enhanced Binary Search Algorithms are a well-known tool in areas such as programming, computational complexity analysis, efficient database searches, etc. We will just describe it briefly. To carry out the search, it is assumed that the elements must be ordered. It consists of dividing the search interval into two parts, comparing the element sought with the central one, in case they are not equal, the extremes of the interval are redefined (depending on whether the central element is greater or less than the one sought) reducing the search space. The process ends when the element is found or, if not found, the most restricted subinterval within the initial search interval, to which the element sought belongs, is returned.

The idea is to compare the searched number a with the element in the middle of the list where the search will be performed. If equal then we find the element, if not, when a is less than the middle element the same strategy is applied to the list to the left of the middle element. And if a is greater than the element in the middle, the same strategy is applied to the list to the right of said element.

Revista Cubana de Ciencias Informáticas
Vol. 17, No. 1, Mes Enero-Marzo, 2023
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 16-30

In the Maple 17 program, it is not necessary to implement this algorithm, since it has several built-in functions that allow binary search to be performed. These include *BinarySearch* and *BinaryPlace*, both from the List-Tools package. As well as the *sort* function for sorting lists. To obtain the n sought that meets the condition, $s_n < [x\mathbf{M}]_{dec} \leq s_{n+1}$, the function *BinaryPlace*.

Note that in essence, H_3 is looking for the smallest decimal distance. In this case, s_n and s_{n+1} are the two closest elements to $[x\mathbf{M}]_{dec}$. If, $[x\mathbf{M}]_{dec} = s_{n+1}$, then $x\mathbf{M} \in S$, otherwise takes the smallest distance between $[x\mathbf{M}]_{dec}$ and the closest element of s_n and s_{n+1}. The main advantage is that it is not necessary to iterate through all the elements of S exhaustively. The operations to obtain S' are performed in the pre-calculus stage.

Experiments and results

The GA was applied to the same 20 systems in previous experiments, under the same conditions, but using the H_3 function. As result, 3.7 solutions were found on average, in one generation in all systems and in a time of approximately 0.448 seconds. Note that the number of generations is less than that obtained with H_1 and H_2. The number of solutions is very similar in all three cases, and greater than with H_2. The interesting thing is the decrease in the time consumed, less than with the two previous functions, and in no case did it exceed 0.5 seconds.

Other experiments with larger systems are described below, with the aim of comparing the functions H_1 and H_3, these two being the fastest. In all cases, the GA parameters are the same as those in previous experiments in this paper. 40 MRHS equations systems were randomly generated, with $n = 16$ variables and $m = 4$ MRHS equations each. The sets S_1, S_2, S_3 and S_4 each have eight elements, so S has 4096. The 40 systems were divided into two groups of 20 systems each: GI and GII. In the GI group the four blocks of $\mathbf{M}$ have four columns, so $\mathbf{M}$ has 16 columns in total. In group GII, $\mathbf{M}$ has 24 columns. In other words, increasing the number of columns in $\mathbf{M}$ increases the size of the space that contains the total number of elements that the set S can take. In this sense, the more the columns of $\mathbf{M}$ increase, keeping the other parameters fixed and especially the cardinal of S, the more difficult it is to find a solution for the MRHS system.

The results for the GI group are shown in Table 1. The second column is the number of solutions per system; the third is the number of generations in which those solutions were found, and the fourth column is the average time, in seconds, that the GA took in that number of generations.

Table 1 - Results obtained with the GI group systems

Function	Solutions	Gen.	Time (s)
H_1	8.4	1	15.482
H_3	8.4	1	0.515

Note that the two functions had the same results in terms of number of solutions and generations. However, there is a big difference in the time consumed, with H_3 being on average 30 times faster than H_1. The results for group GII are shown in Table 2. In this case the times are given in minutes. Unlike Table 1, another column is added, in which the number of systems for which the GA did not find a solution is reflected. In the cases where no solution was found, and therefore the GA went through the 81 generations, the times were approximately 13.7 and 0.42 minutes for H_1 and H_2, respectively. The H_3 function was more effective in the four comparison criteria, but above all, the main difference is again in relation to time. In this sense, H_3 never reached 0.5 minutes (30 seconds), when H_1 needed more than 10 minutes on average.

Table 2 - Results obtained with the GII group systems

Function	Solutions	Gen.	Time (min)	Faults
H_1	0.5	59.05	10.023	10
H_3	0.55	52.05	0.2726	9

The two functions were tested with a much larger random MRHS system than the previous ones. The system has $n = 64$ variables and $m = 4$ equations. $\mathbf{M}$ has a total of 32 columns, and S has 65536 elements. The H_3 function found a solution, and the total time to traverse all 81 generations was approximately 1.15 minutes. In contrast, with H_2 the search was interrupted after eight hours of compilation.

In summary, it is interesting to note that with the H_3 function methodology, it is possible to considerably reduce the time consumed by the GA, maintaining the same level of effectiveness in terms of the number of solutions, generations, and times when no solutions are found (failures). In other words, you gain speed without sacrificing any of the other criteria.

Conclusions

MRHS systems constitute a novel alternative method for algebraic work with block ciphers within Algebraic Cryptanalysis. The proposal of GAs as a solution method for MRHS systems is still a young line, but with promising results. With the experiments carried out and the fitness functions proposed, it is shown that the GA not only solves the MRHS systems, but that it is capable of finding more than one solution. With one of the fitness functions proposed, H_3, it is possible to considerably reduce the time consumed by the GA in the search for solutions. In future research, we will work on modeling specific block ciphers as MRHS equations systems, and then carry out attacks on the encryption looking for the key as a solution of the MRHS system through the GA.

References

S. Aashiqbanu, B. Krishna Murthy, G. Bindu Sai, Gali Sowmya, Kalluru Hemaswitha, and Rengarajan Amirtharajan. Telugu DNA for Safe Delivery: A Secured Text Communication. *Wireless Personal Communications*, 2022. URL https://doi.org/10.1007/s11277-022-09901-w.

Riyadh Bassil Abduljabbar, Oday Kamil Hamid, and Nazar Jabbar Alhyani. Features of genetic algorithm for plain text encryption. *International Journal of Electrical and Computer Engineering*, 11(1):434–441, 2021. URL https://doi.org/10.11591/ijece.v11i1.pp434-441.

Firat Artuğer and Fatih Özkaynak. SBOX-CGA: substitution box generator based on chaos and genetic algorithm. *Neural Computing and Applications*, 2022. URL https://doi.org/10.1007/s00521-022-07589-4.

Miguel Borges-Trenard, Mijail Borges-Quintana, and Lázaro Monier-Columbié. An application of genetic algorithm to cryptanalysis of block ciphers by partitioning the key space. *J. Discret. Math. Sci. Cryptogr.*, 2019. URL https://doi.org/10.1080/09720529.2019.1649028.

Chunping Cao, Zhe Cen, Xiutao Feng, Zhangyi Wang, and Yamin Zhu. Straightforward Guess and Determine Analysis Based on Genetic Algorithm. *Journal of Systems Science and Complexity*, 2022. URL https://doi.org/10.1007/s11424-022-1031-x.

J. P Indrøy. Algebraic Attack on Small Scale Variants of AES using Compressed Right Hand Sides. *Thesis of Grade, Secure and Reliable Communications, University of Bergen*, 2018.

J. P. Indrøy and H Raddum. Trail Search with CRHS Equations. Cryptology ePrint Archive, Paper 2021/1329, 2021. URL https://eprint.iacr.org/2021/1329.

Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, 2021. URL https://doi.org/10.1007/s11042-020-10139-6.

K. Matheis, R. Steinwandt, and A Suárez Corona. Algebraic Properties of the Block Cipher DESL. *Symmetry*, 11(1411), 2019. URL https://doi.org/10.3390/sym11111411.

Ayush Mittal and Ravindra Kumar Gupta. Encryption and Decryption of a Message Involving Genetic Algorithm. *International Journal of Engineering and Advanced Technology*, 9(2):3920–3923, 2019. URL https://doi.org/10.35940/ijeat.B2379.129219.

Ramzi A Nofal. Analyzing and Reducing the Overhead of TLS and DTLS in WiFi-Based IoT Networks. *Engineering Ph.D. Theses*, 41, 2022. URL https://scholarcommons.scu.edu/eng_phd_theses/41.

Tingxiu Qiu and Qichun Wang. A Search Cryptographically Significant S-Boxes with Improved DPA Resistance Based on Genetic Algorithm. In Sun, X., Zhang, X., Xia, Z., Bertino, E. Advances in Artificial Intelligence and Security, editor, *Communications in Computer and Information Science*, volume 1588. Springer, Cham, 2022. ISBN 978-3-031-06764-8. URL https://doi.org/10.1007/978-3-031-06764-8_7.

H. Raddum and I Semaev. New technique for solving sparse equation systems. *Ecrypt's STVL website, January 16th 2006, see also Cryptology ePrint Archive, 2006/475*, 2006.

H. Raddum and I Semaev. Solving Multiple Right Hand Sides linear equations. *Des. Codes Cryptography*, 49(1):147–160, 2008. URL https://doi.org/10.1007/s10623-008-9180-z.

H. Raddum and P Zajac. MRHS solver based on linear algebra and exhaustive search. *J. Math. Cryptol*, 12: 143–157, 2018.

O. Tito-Corrioso, M.A. Borges-Trenard, M. Borges-Quintana, O. Rojas, and G. Sosa-Gómez. Study of Parameters in the Genetic Algorithm for the Attack on Block Ciphers. *Symmetry*, 13(806), 2021. URL https://doi.org/10.3390/sym13050806.

Osmani Tito-Corrioso, Miguel A Borges-Trenard, and Mijail Borges-Quintana. Ataques a cifrados en bloques mediante búsquedas en grupos cocientes de las claves. *Ciencias Matemáticas*, 33(1):71–74, 2019.

Pavol Zajac. On the feasibility of algebraic cryptanalysis by bit flipping. In *Book of Abstracts, 21th Central European Conference on Cryptology*, pages 67–68. Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2021.

Pavol Zajac and Peter Špaček. A New Type of Signature Scheme Derived from a MRHS Representation of a Symmetric Cipher. *Infocommunications Journal*, XI(4):23–30, 2019. URL https://doi.org/10.36244/ICJ.2019.4.4.

Conflicts of Interest

The authors declare no conflict of interest and authorize the distribution and the use of this paper.

Author Contributions

1. Conceptualization: Miguel A. Borges-Trenard
2. Data's healing: Osmani Tito-Corrioso
3. Formal Analysis: Osmani Tito-Corrioso, Mijail Borges-Quintana
4. Funding: Osmani Tito-Corrioso, Mijail Borges-Quintana and Miguel A. Borges-Trenard
5. Investigation: Osmani Tito-Corrioso, Mijail Borges-Quintana and Miguel A. Borges-Trenard
6. Methodology: Osmani Tito-Corrioso, Mijail Borges-Quintana and Miguel A. Borges-Trenard
7. Project's Administration: Mijail Borges-Quintana

8. Resources: Osmani Tito-Corrioso, Mijail Borges-Quintana and Miguel A. Borges-Trenard
9. Software: Osmani Tito-Corrioso
10. Supervision: Mijail Borges-Quintana and Miguel A. Borges-Trenard
11. Validation: Mijail Borges-Quintana and Miguel A. Borges-Trenard
12. Visualization: Osmani Tito-Corrioso and Mijail Borges-Quintana
13. Writing - original rough draft: Osmani Tito-Corrioso
14. Writing - checking and edition: Osmani Tito-Corrioso, Mijail Borges-Quintana and Miguel A. Borges-Trenard

Funding

The research associated with the results presented in this publication received funds from the International Funds and Projects Management Office under the code PN223LH010-024; and also from Red CYTED "NUEVAS HERRAMIENTAS CRIPTOGRÁFICAS PARA LA E-COMUNIDAD".