

Searching for an Efficient System of Equations Defining the AES Sbox for the QUBO Problem

Elżbieta Burek¹, Krzysztof Mańk¹, and Michał Wroński²

¹*Military University of Technology, Warsaw, Poland,*

²*NASK National Research Institute, Warsaw, Poland*

<https://doi.org/10.26636/jtit.2023.4.1340>

Abstract — The time complexity of solving the QUBO problem depends mainly on the number of logical variables in the problem. This paper focuses mainly on finding a system of equations that uniquely defines the Sbox of the AES cipher and simultaneously allows us to obtain the smallest known optimization problem in the QUBO form for the algebraic attack on the AES cipher. A novel method of searching for an efficient system of equations using linear-feedback shift registers has been presented in order to perform that task efficiently. Transformation of the AES cipher to the QUBO problem, using the identified efficient system, is presented in this paper as well. This method allows us to reduce the target QUBO problem for AES-128 by almost 500 logical variables, compared to our previous results, and allows us to perform the algebraic attack using quantum annealing four times faster.

Keywords — AES Sbox, cryptanalysis, minimal equation system for Sbox, quantum annealing, QUBO

1. Introduction

Nowadays, cryptography is relied upon in almost all spheres of life and serves as a basis of security of communication processes, increasingly ensuring the security of private and business data as well. In today's computerized world, it is relatively difficult to eliminate an existing vulnerable protocol from use, because the process involves upgrading all devices that rely on the specific solution. Therefore, the task may take over a decade to complete. Many critical components of the cybersecurity infrastructure used in the public domain and in industry have remained unchanged for years. Many devices that are in use currently or are scheduled to be implemented soon will have to operate for the next few decades with minimal change. Therefore, it is important to respond to all potential threats and carefully manage the risks associated with potentially successful attacks of various types.

In recent years, quantum computing has challenged existing cryptographic approaches, as the security of encryption and authentication schemes is based on mathematical problems that are difficult to solve using classical computers, but their solutions are easy to check. These safety standards have worked well for decades, because there were no exploits to

break them. Over time, it has become common to assume that if a problem cannot be solved using bits, it cannot be solved at all.

However, the continuous development of quantum computers, representing a completely new paradigm of computation, putting aside bits in favor of qubits, has revived interest in the potential of this technology. Since the implementation of modern infrastructure is time consuming, regardless of whether we can estimate the exact time required to build a sufficiently large quantum computer, information security systems should be prepared now to resist attacks conducted with the use of such a computer. The development of current quantum computers is based on two main approaches.

The first approach, called gate-based quantum computing, is similar to today's classic computation, where the problem is presented as a sequence of basic operations (gates) used to manipulate the state of qubits. The other approach, known as adiabatic quantum computing, consists of smooth changes of the system's Hamiltonian, from the initial state to the final Hamiltonian in which the problem is encoded. If the system's evolution is slow enough to remain in the ground state of the changing Hamiltonian, then the final state corresponds to the problem's solution. In the real world, this requirement is relaxed, and the protocol implementing this computational model is called quantum annealing.

The main difference between the two models is that quantum annealing solves optimization problems, while gate-based quantum computation can be used for a wider range of problems. Therefore, the number of quantum attacks using a gate-based quantum computer is growing faster than the number of those relying on quantum annealing. On the other hand, gate-based quantum computation is more sensitive to noise and quantum computational errors, creating a large overhead in the number of qubits and running time.

In contrast, quantum annealing is considered relatively immune to some errors, such as noise and decoherence. Many companies, such as IBM, Google, D-Wave Systems, Cambridge Quantum Computing or Rigetti, are working on building such a machine. Currently, the largest quantum computer is the D-Wave Advantage, with 5,760 physical qubits. It uses the quantum annealing method and can already compete with classical units when it comes to solving some problems.

The potential of quantum computers was presented in 1994 by Peter Shor [1], who developed quantum algorithms to solve the factorization problem and the discrete logarithm problem in polynomial time. Then, Bennett *et al.* [2] proved that quantum algorithms that solve problems using the black box model, i.e., ignoring the detailed structure of the problem, must perform at least $N^{1/2}$ steps, where N is the size of the problem. In [3], Grover presented a quantum algorithm that finds such a solution in $O(N^{1/2})$ steps. Grover's algorithm is not as destructive to symmetric cryptography as Shor's algorithm is to asymmetric cryptography, but it significantly degrades the security of ciphers.

These algorithms were developed for gate-based quantum computers, and their deployment would lead to a huge advantage in terms of information security. However, for the size of the problems we are interested in, these algorithms require several orders of magnitude more resources than current gate-based quantum computers offer. However, even if there were a quantum computer capable of running Grover's algorithm, the solution to the problem of cipher security is simple – it is sufficient to increase the key size. But what if a sophisticated quantum attack was experienced using the detailed structure of the problem that was faster than a brute-force attack? This is an open question, and further research is needed in this area.

Algebraic attacks using quantum annealing may serve as an example of quantum attacks using relying on the cipher structure. They are based on algebraic attacks, representing the cipher by means of a polynomial equations system which must be solved. If the problem of solving such a system of equations is presented as an appropriate optimization problem, then the quantum annealing method may be harnessed to find the solution. In [4], we introduced a method of transforming a polynomial equations system into an optimization problem in QUBO, acceptable by the D-Wave quantum computer for the AES cipher. We have shown that, unlike in the case of classic tools used for algebraic attacks, we do not need to build over-defined systems of equations. Moreover, for the specific instance of the AES cipher, using the D-Wave quantum annealer, we also showed that the proposed method allows for the recovery of the correct key. Therefore, the next step is to reduce the size of the QUBO problem for the AES cipher.

This article presents a further search for an efficient equations system defining the Sbox of the AES cipher, which was initiated in our previous paper [4].

2. Impact of Transformation on the Size of the QUBO Problem

2.1. Proposed Transformation to the QUBO Problem

Let the system of multivariate polynomial equations f_i over $GF(2)$ be given, which describes the block cipher. The proposed method of transformation is performed in the following steps.

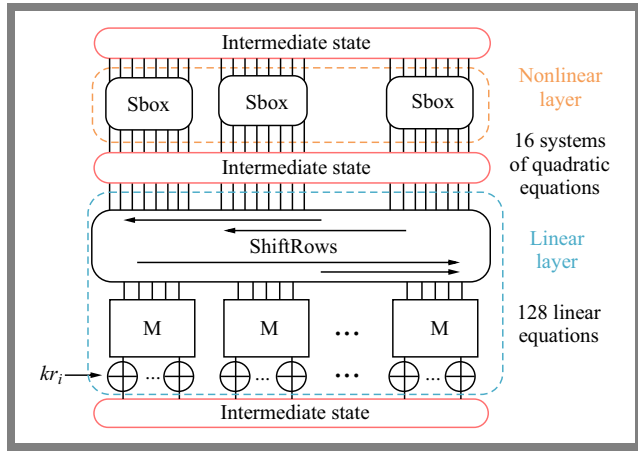


Fig. 1. Splitting the AES cipher round using additional variables.

Each equation f_i is transformed into pseudo-Boolean functions of the $f'_i = f_i - 2 \cdot k_i$ form, where k_i is an integer and $k_i \leq \lfloor \frac{f_{i,max}}{2} \rfloor$ and the maximal value of polynomial $f_{i,max}$ is a value of polynomial f_i when all of its binary variables are equal to one.

The value of k_i has to be written as the sum of Boolean variables k_{ij} , according to:

$$k_i = \sum_{j=0}^{bl(k_{i,max})-2} 2^j k_{ij} + (k_{i,max} - 2^{bl(k_{i,max})-1} + 1) \cdot k_{i,bl(k_{i,max})-1}, \quad (1)$$

where $bl(x)$ is the bit-length of integer x and

$$k_{i,max} = \lfloor \frac{f_{i,max}}{2} \rfloor.$$

Each equation f'_i is linearized using a linearization with a penalty. We used the Rosenberg linearization [5], where each quadratic monomial is replaced by a new auxiliary binary variable as:

$$x_i x_j \rightarrow x_k + 2(x_i x_j - 2x_k(x_i + x_j) + 3x_k), \quad (2)$$

where $2(x_i x_j - 2x_k(x_i + x_j) + 3x_k)$ is a penalty and is added to the cost function. If $x_i x_j$ equals x_k , then the penalty is zero. Otherwise, the penalty is a positive integer and this solution is rejected.

The last step is to find the sum of squares of all polynomials f'_i , obtaining $\sum_i (f'_{i,lin} - 2 \cdot k_i)^2$, where $f'_{i,lin}$ is the f_i equation after linearization.

Finally, the problem in the QUBO form is obtained as:

$$\sum_i (f'_{i,lin} - 2 \cdot k_i)^2 + M \cdot Pen_{lin} - C, \quad (3)$$

where C is a constant appearing in the polynomial:

$$\sum_i (f'_{i,lin} - 2 \cdot k_i)^2 + M \cdot Pen_{lin},$$

Pen_{lin} is a penalty after all substitutions and M is a positive constant.

A more detailed description of the transformation method may be found in [4].

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
x_1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
x_3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
x_0	1	0	1	1	1	0	1	0	0	0	0	0	1	1	0
y_1	0	1	0	0	1	0	0	1	1	0	0	0	1	1	1
y_2	0	0	1	1	0	0	0	0	1	1	0	1	0	1	1
y_3	1	0	0	1	1	1	0	1	0	0	0	1	0	0	1
$x_0 y_1$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
$x_0 y_2$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
$x_0 y_3$	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
$x_1 y_2$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
$x_1 y_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
$x_2 y_3$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
$x_0 y_0$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
$x_0 y_1$	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1
$x_0 y_2$	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1
$x_0 y_3$	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
$x_1 y_0$	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0
$x_1 y_1$	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1
$x_1 y_2$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
$x_1 y_3$	0	0	0	0	1	1	0	1	0	0	0	0	0	1	1
• • •															
$x_1 y_3$	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1
$x_2 y_3$	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1

Fig. 2. A fragment of the matrix of monomial values for all possible inputs of the example Sbox.

2.2. Impact of the AES Cipher Transformation on the Problem Size

In the presented transformation method, two steps affect the size of the final QUBO problem:

- linearization, where each quadratic monomial is a new, auxiliary binary variable,
- determination of the value of multiples of k_i , where the number of auxiliary binary variables for a given equation f_i increases logarithmically with the number of all monomials in that equation.

To obtain equations of a degree of at most two, describing the AES cipher, intermediate states between the linear and non-linear layers of the cipher were introduced using additional binary variables, as shown in Fig. 1. As a result, the linear layer is represented by 128 linear equations that uniquely determine the operations of the linear layer. Furthermore, the non-linear layer is represented by 16 instances of a quadratic equations system which defines the Sbox of the AES cipher. So, if we want to reduce the size of the QUBO problem, we need to find an efficient system of equations defining the Sbox.

The Sbox of an AES cipher can be defined as a bijection: $Sbox : F_{2^8} \rightarrow F_{2^8}$ and can be represented by implicit equations in the following form:

$$f_i(x_0, \dots, x_7, y_0, \dots, y_7) = 0, \tag{4}$$

where x_0 to x_7 are inputs to the Sbox and y_0 to y_7 are outputs, and the following implication holds:

$$Sbox(x_0, \dots, x_7) = y_0, \dots, y_7 \Rightarrow f_i(x_0, \dots, x_7, y_0, \dots, y_7) = 0. \tag{5}$$

Courtois and Pieprzyk showed, in [6], how to generate an overdefined system of implicit multivariate equations of a degree of at most two for the Rijndael Sbox. Let us consider the following example. Let the Sbox be defined by the following permutation:

$$(9, 4, 10, 11, 13, 1, 8, 5, 6, 2, 0, 3, 12, 14, 15, 7). \tag{6}$$

This Sbox size is 4×4 , so there are 16 possible inputs and 37 possible monomials of degree two or less. To determine polynomial equations for this Sbox, a matrix with the dimensions of 37×16 is created, with each row containing the values of a given monomial for all possible inputs. A fragment of this matrix is shown in Fig. 2. By performing Gaussian elimination and storing the operations performed on the rows, we obtain equations that satisfy the Sbox.

For example, XOR-ing the gray rows in Fig. 2 yields zero, so the equation:

$$x_0 + x_0x_1 + x_0x_2 + x_0y_2 + x_0y_3 + x_1y_2 = 0,$$

consisting of the monomials of those rows satisfies Sbox. Performing the method described above for the Sbox of the AES cipher, for which there are 256 possible inputs and 137 possible monomials of degree two or less, 39 quadratic equations are obtained by Gaussian elimination. We asked whether all the 39 equations are necessary to determine the Sbox of the AES cipher unambiguously.

3. Searching for an Efficient System of Equations Defining the AES Sbox

3.1. Definition of the Problem of Finding an Efficient System of Equations

The problem of finding an efficient system describing the Sbox can be defined as an optimization problem, where the solution space is the set of all systems of quadratic equations uniquely defining the Sbox, and the objective function assigns, to each system, the number of additional binary variables needed to perform the transformation to the QUBO problem. Therefore, the minimum solution to the problem defined in this way is a system with the following properties:

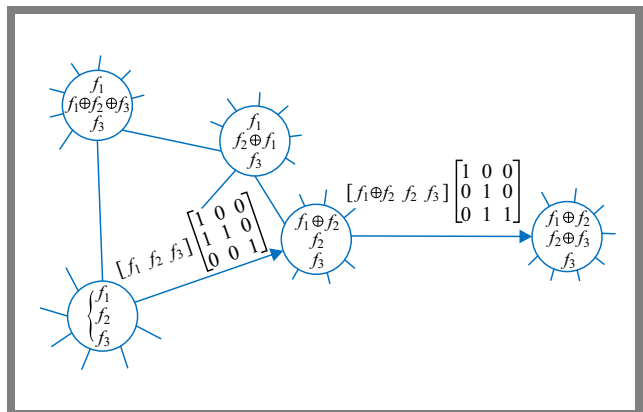


Fig. 3. Fragment of a graph representing a set of derived systems for the initial system of three equations.

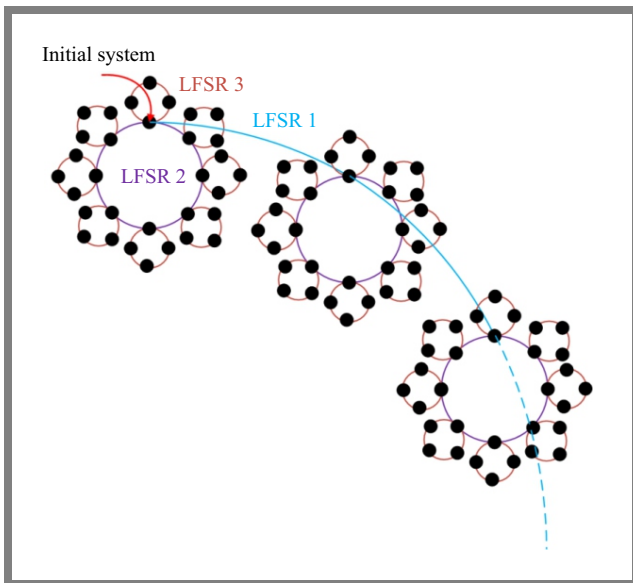


Fig. 4. General outline of the process of searching for an efficient system using method 1.

Tab. 1. The number of systems uniquely defining the AES Sbox for at most 14 equations.

Number of equations	Number of systems
<12	0
12	1,052
13	2,690,682
14	227,550,310

- due to linearization, the number of different quadratic monomials in the system is as small as possible,
- due to the determination of the value of multiples of k_i , the equations system should mostly consist of very short equations and a few longer ones. In the ideal scenario, the system should consist of one equation with a large number of monomials and the remaining equations with two monomials each.

To determine the space of solutions, the following question must be answered: when does the equations system uniquely define the Sbox of the AES cipher? The answer is: a given system consisting of a number of equations, selected from all 39 quadratic equations, uniquely defines the Sbox of the AES cipher if, for each of the 256 possible inputs, all equations of this system are satisfied for exactly one output. We checked all subsets consisting of 14 quadratic equations at the most. The maximum size of the subset was assumed to be 14 due to computational resources and available time. The number of the existing equations systems uniquely defining the AES Sbox is presented in Tab. 1.

According to the definition of the problem of finding an efficient equations system, the minimum solution depends on the number of different quadratic monomials in the system and the number of all monomials in each equation. Each equation system in Tab. 1 consists of a number of different quadratic monomials which cannot change. However, we can reduce the

number of all monomials in the equations of a given system by applying the XOR operation.

Let F denote the set of polynomial equations of the analyzed system and let G denote a subset of F such that:

$$|G| = r, r \in \{2, \dots, |F| - 1\}$$

and

$$G = \{g_j(x_0, \dots, x_7, y_0, \dots, y_7) : g_j \in F \wedge (i \neq j \implies g_i \neq g_j)\}, \text{ for } i = \overline{0, r-1}.$$

If the number of monomials of the $h = g_0 \oplus g_1 \oplus \dots \oplus g_{r-1}$ polynomial is less than the number of monomials of any of the g_0, \dots, g_{r-1} polynomials, then the g_i polynomial with the greatest difference in the number of monomials is replaced by the h polynomial.

This substitution preserves the uniqueness of defining the Sbox. Suppose that for a given input/output pair, all g_i polynomials for $i = \overline{0, r-1}$ are 0. Then $h = 0$ and substituting it for any of the g_i polynomials does not affect the satisfiability of the system, because it depends on the other polynomials of the set F . However, suppose at least one g_i polynomial has the value of 1. In that case, regardless of the values of the other polynomials of the set F , the system should not be satisfied for this input/output pair. Therefore, let us consider two cases:

- 1) The h polynomial has the value of 1, i.e., an odd number of g_i polynomials from the set G has the value 1, then replacing any of the polynomials $g_i = 1$ or $g_i = 0$ with the $h = 1$ polynomial does not change the decision about the unsatisfiability of the system for this input/output pair. At most, the number of polynomials not satisfying the Sbox for this input/output pair will increase by one.
- 2) The h polynomial has the value of 0, i.e., an even number of g_i polynomials from the set G has the value 1, then replacing any of the polynomials $g_i = 1$ or $g_i = 0$ with the $h = 0$ polynomial does not change the decision about the unsatisfiability of the system for this input/output pair, because among the remaining g_i polynomials, there is at least one more polynomial not satisfying the Sbox for that input/output pair.

In addition, this substitution does not change the number of equations in the system and does not change the number of different quadratic monomials in the system. Therefore, for each equations system in Tab. 1 (let us call it the initial system), there are a number of derivative systems, also uniquely defining the Sbox of the AES cipher and resulting from the XOR operation on the initial system's equations. The set of such derivative systems can be represented as a complete graph whose vertices are the systems obtained after performing the XOR operation on a certain number of equations. Presenting the system as a vector, where the elements of this vector are the equations of the system, a single operation of XOR-ing the equations can be defined as multiplying the vector by an invertible binary matrix. Hence, the number of all possible derivative systems for a given initial system is equal to the number of all invertible binary matrices divided by the

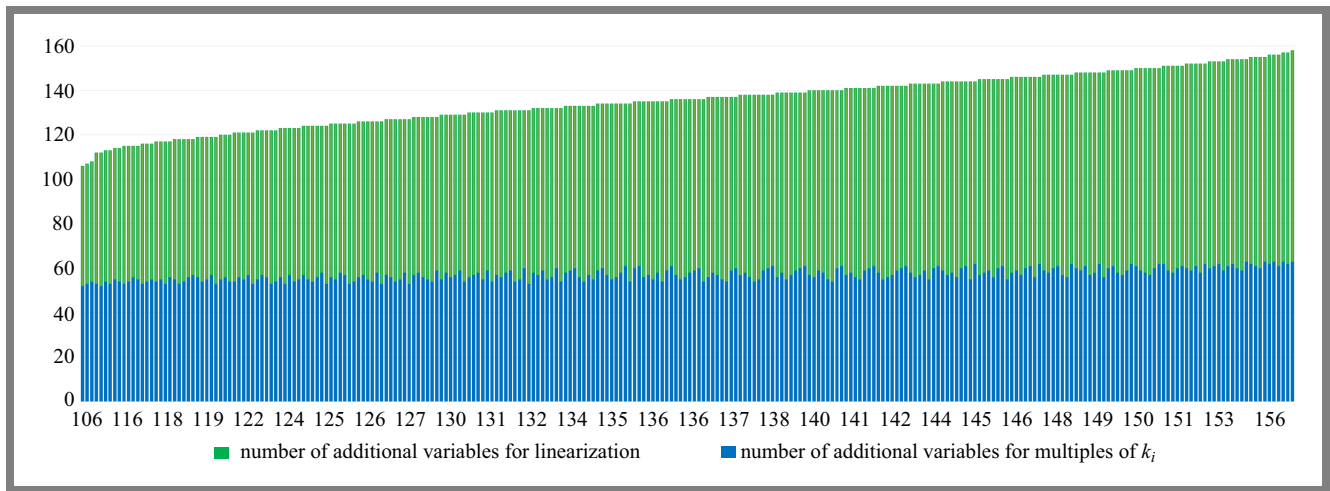


Fig. 5. The number of additional binary variables needed to transform a system of thirteen equations into the target QUBO problem.

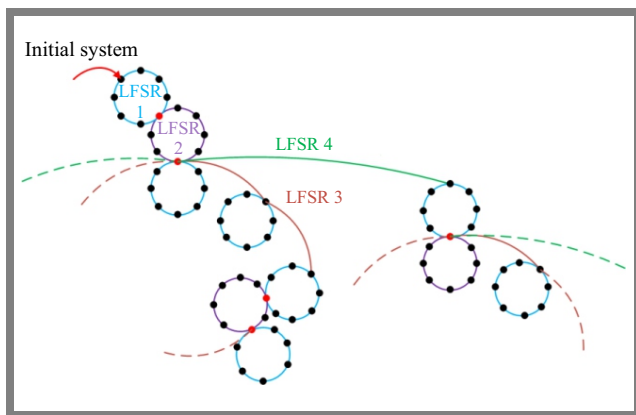


Fig. 6. General outline of the process of searching for an efficient system using method 2.

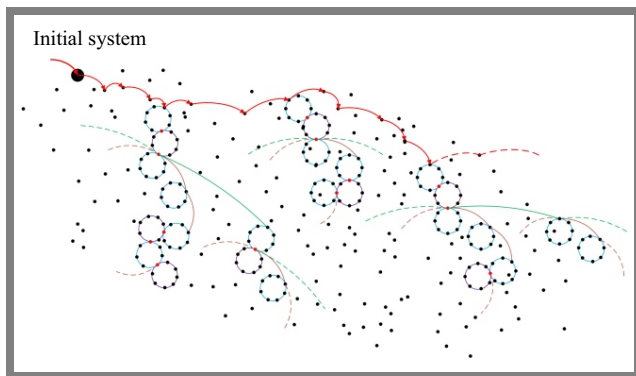


Fig. 7. General outline of the process of searching for an efficient system using method 3.

number of permutations of the equations:

$$\frac{\prod_{i=0}^{n-1} 2^n - 2^i}{n!}, \tag{7}$$

where n is the number of equations in the initial system. Figure 3 shows a fragment of the graph representing a set of derivative systems for a certain initial system consisting of 3 equations. A complete graph consists of 28 vertices. For the Sbox of the AES cipher for each of the 1,052 systems of twelve equations, there are $1.3 \cdot 10^{34} \approx 2^{114}$ possible derivative

systems (vertices in the graph). For each of the 2,690,682 systems of thirteen equations, there are $3.5 \cdot 10^{40} \approx 2^{135}$ systems and for each of the 227,550,310 systems of fourteen equations, there are $3.3 \cdot 10^{47} \approx 2^{158}$ systems. Of course, it is not possible to search such large spaces, so certain restrictions have been adopted:

- to effectively perform the multiplication operation, we used linear feedback shift registers, where the vector representing the equations system is the state of LFSR,
- each LFSR was defined with an arbitrarily chosen primary polynomial to obtain the full period,
- for a given initial system, we assumed a time horizon of 20 minutes.

3.2. Searching for an Efficient System – Method 1

The first method is analogous to the full search. The number of binary variables needed to represent the value of multiples of k_i was adopted to measure the system’s efficiency. The idea behind the developed method is shown in Fig. 4, where black circles represent systems of equations. The search process starts in a given initial system, and next, the equations are XOR-ed according to the polynomial of LFSR3. In each LFSR3 state, we check whether the resulting new equations system is more efficient than the one found. When LFSR3 completes a full cycle, we execute one step of LFSR2 and perform the search again with LFSR3. If LFSR2 completes a full cycle, one step of LFSR1 is executed, and we perform the search again with LFSR3 and LFSR2.

All initial systems of twelve equations and approximately 10% of initial systems of thirteen and fourteen equations from Tab. 1 were taken for the search with method 1. In 20 min, we checked approximately $8.6 \cdot 10^{10}$ derived systems for a given initial system of equations.

In Fig. 5, the influence of the linearization process and the process of representing the multiples of k_i using variables on the size of the final problem in the form of QUBO is presented. A single column in this graph represents the total number of additional binary variables needed to transform a system

Tab. 2. Number of additional binary variables to transform the nonlinear layer of the AES cipher to the QUBO problem.

System of Sbox	For one Sbox	For nonlinear layer
39 equations	299	4, 784
Paper [4]	106	1, 696
This paper	105	1, 680

of thirteen equations into a QUBO problem. The blue part of this column shows the number of additional variables used to represent multiples of k_i , while the green part shows the number of additional variables for the linearization process. As one may notice, the cost of linearization has a greater impact on the size of the target QUBO problem. The most efficient system of equations of all tested systems consists of thirteen equations and 54 different quadratic monomials, and its transformation to the QUBO problem requires 106 additional binary variables.

3.3. Searching for an Efficient System – Method 2

In the second method, the value determining the direction of the move performed within the graph was assumed. According to the definition of the objective function of the problem of finding an efficient system presented earlier, the difference between the number of all monomials in the equations must be as large as possible. Therefore, we assumed that the variance of the number of all monomials in the equations of the system determines the direction of moving to the next vertex of the graph. Furthermore, as in the previous method, the number of variables needed to represent the multiples of k_i of all system equations is the measure of the system’s efficiency.

The equations system’s search space is shown in Fig. 6. The search process starts with the initial system. In each state of LFSR1 and LFSR2, the variance of the number of all monomials in the system’s equations and the number of binary variables for the value of multiples of k_i are checked. In Fig. 6, systems with the maximum variance in a given cycle are marked with red dots. If a full cycle of LFSR1 is performed and a new equations system with the maximum variance is found, then it is the initial state of LFSR2. The transition to subsequent equations systems, alternately by means of LFSR1 and LFSR2, is carried out until one of these registers returns to its initial state without finding a new system with the maximum variance in its cycle. Then, LFSR3 is used to exit this state by moving to another vertex of the graph, and the search is started again using LFSR1 and LFSR2 alternately. If LFSR3 completes a full cycle, a move to another vertex of the graph is performed by LFSR4, and the procedure with three registers is repeated. The search ends when the fourth register completes a full cycle.

We did not find a more efficient system than the one identified with the use of the first method.

Tab. 3. Results of transformation of the system of multivariate quadratic equations describing the AES to the QUBO problem.

Variant of AES	System of Sbox	Size of QUBO
AES-128	39 equations	68,600
	Paper [4]	30,026
	This paper	29,528
AES-192	39 equations	138,632
	Paper [4]	58,920
	This paper	57,384
AES-256	39 equations	165,731
	Paper [4]	70,059
	This paper	68,187

3.4. Searching for an Efficient System – Method 3

In the third method, we tried to increase the range of the search space. Therefore, a given initial system is first multiplied many times (from 1 to 1335) by a random binary matrix, creating new initial systems for the search. The search process itself is carried out using method 2. The scheme of the search process relying on method 3 is shown in Fig. 7. As a result of applying this method, a slightly more efficient system was found than in method 1. The obtained efficient system consists of 13 equations and 54 different quadratic monomials, and the cost of its transformation to the QUBO problem is 105 additional binary variables. The polynomials of the obtained equations system are as follows:

$$f_1 = x_0y_6 + x_0y_7 + x_1y_1 + x_1y_7 + x_2y_1 + x_2y_2 + x_2y_3 + x_2y_4 + x_2y_6 + x_3y_0 + x_3y_1 + x_3y_4 + x_3y_7 + x_4y_2 + x_4y_3 + x_4y_4 + x_4y_5 + x_4y_6 + x_5y_0 + x_5y_1 + x_6y_1 + x_6y_2 + x_6y_3 + x_6y_5 + x_6y_6 + x_7y_2 + x_7y_3 + x_7y_5 + x_1 + x_3 + y_5,$$

$$f_2 = x_0y_0 + x_0y_5 + x_0y_7 + x_1y_5 + x_2y_1 + x_2y_2 + x_2y_5 + x_3y_1 + x_3y_4 + x_3y_5 + x_4y_1 + x_4y_3 + x_4y_5 + x_4y_7 + x_5y_1 + x_5y_3 + x_5y_6 + x_5y_7 + x_6y_1 + x_6y_2 + x_6y_4 + x_6y_5 + x_7y_2 + x_7y_3 + x_7y_5 + x_1 + x_2 + x_3 + x_7 + y_3 + y_4,$$

$$f_3 = x_0y_5 + x_0y_6 + x_1y_1 + x_1y_5 + x_2y_0 + x_2y_1 + x_2y_3 + x_2y_4 + x_3y_0 + x_4y_0 + x_4y_1 + x_4y_3 + x_4y_4 + x_4y_7 + x_5y_0 + x_5y_1 + x_5y_3 + x_5y_4 + x_5y_5 + x_5y_7 + x_6y_1 + x_6y_2 + x_7y_2 + x_7y_3 + x_7y_5 + x_0 + x_1 + x_2 + y_0 + y_6 + 1,$$

$$f_4 = x_0y_3 + x_0y_6 + x_0y_7 + x_1y_1 + x_1y_5 + x_2y_0 + x_2y_1 + x_2y_4 + x_2y_6 + x_2y_7 + x_3y_0 + x_3y_2 + x_3y_4 + x_3y_6 + x_4y_3 + x_4y_7 + x_5y_0 + x_5y_4 + x_6y_1 + x_6y_4 + x_7y_2 + x_7y_3 + x_7y_5 + x_7y_7 + x_0 + x_1 + x_2 + x_6 + y_3 + y_4 + y_5,$$

$$f_5 = x_0y_0 + x_0y_2 + x_0y_3 + x_0y_4 + x_0y_5 + x_0y_7 + x_1y_1 \\ + x_1y_5 + x_2y_3 + x_2y_4 + x_2y_5 + x_2y_6 + x_2y_7 + x_3y_1 \\ + x_3y_7 + x_4y_1 + x_4y_5 + x_5y_1 + x_5y_6 + x_5y_7 + x_6y_2 \\ + x_6y_3 + x_6y_6 + x_7y_5 + x_3 + x_4 + x_7 + y_0 + y_5 \\ + y_7 + 1,$$

$$f_6 = x_1y_1 + x_2y_0 + x_2y_3 + x_2y_7 + x_3y_2 + x_3y_5 + x_3y_7 \\ + x_4y_0 + x_4y_2 + x_4y_4 + x_4y_7 + x_5y_1 + x_5y_2 + x_5y_4 \\ + x_5y_5 + x_5y_6 + x_6y_0 + x_6y_1 + x_6y_4 + x_6y_5 + x_7y_1 \\ + x_7y_2 + x_7y_3 + x_7y_5 + x_1 + x_6 + y_2 + y_5 + y_7$$

$$f_7 = x_0y_3 + x_0y_4 + x_0y_5 + x_1y_1 + x_1y_5 + x_2y_4 + x_2y_6 \\ + x_3y_0 + x_3y_1 + x_3y_6 + x_3y_7 + x_4y_0 + x_4y_1 + x_4y_3 \\ + x_4y_6 + x_4y_7 + x_5y_2 + x_5y_3 + x_5y_6 + x_6y_2 + x_6y_5 \\ + x_7y_1 + x_7y_7 + x_0 + x_4 + x_5 + y_0 + y_4 + y_6 + y_7,$$

$$f_8 = x_0y_0 + x_0y_4 + x_0y_6 + x_1y_1 + x_1y_5 + x_2y_0 + x_2y_1 \\ + x_2y_2 + x_2y_3 + x_2y_4 + x_2y_5 + x_2y_6 + x_2y_7 + x_3y_0 \\ + x_3y_5 + x_4y_7 + x_5y_0 + x_5y_6 + x_6y_5 + x_7y_2 + x_7y_7 \\ + x_0 + x_5 + x_6 + x_7 + y_3 + y_4 + y_5 + y_6 + 1,$$

$$f_9 = x_0y_7 + x_1y_1 + x_1y_5 + x_2y_2 + x_2y_4 + x_2y_5 + x_2y_7 \\ + x_3y_3 + x_3y_4 + x_4y_2 + x_4y_4 + x_4y_5 + x_5y_0 + x_5y_1 \\ + x_5y_4 + x_6y_0 + x_6y_2 + x_6y_3 + x_7y_2 + x_7y_5 + x_0 \\ + x_1 + x_2 + x_3 + x_6 + x_7 + y_0 + y_2 + y_3 + 1,$$

$$f_{10} = x_0y_0 + x_0y_2 + x_0y_3 + x_0y_5 + x_0y_6 + x_2y_0 + x_2y_1 \\ + x_2y_4 + x_2y_6 + x_3y_1 + x_3y_3 + x_3y_6 + x_3y_7 + x_4y_0 \\ + x_4y_1 + x_4y_6 + x_5y_1 + x_5y_2 + x_5y_4 + x_5y_6 + x_5y_7 \\ + x_6y_2 + x_6y_5 + x_7y_3 + x_7y_7 + x_5 + x_7,$$

$$f_{11} = x_0y_0 + x_0y_2 + x_0y_3 + x_0y_5 + x_1y_1 + x_1y_7 + x_2y_2 \\ + x_2y_7 + x_3y_2 + x_3y_4 + x_3y_5 + x_3y_7 + x_4y_4 + x_4y_5 \\ + x_5y_1 + x_5y_4 + x_5y_6 + x_6y_1 + x_6y_3 + x_7y_7 + x_2 \\ + x_3 + x_6 + y_0 + y_1 + y_5 + y_7,$$

$$f_{12} = x_0y_2 + x_0y_3 + x_0y_6 + x_1y_1 + x_1y_5 + x_1y_7 + x_3y_2 \\ + x_3y_5 + x_3y_7 + x_4y_0 + x_4y_1 + x_4y_2 + x_4y_3 + x_4y_5 \\ + x_5y_0 + x_5y_1 + x_5y_2 + x_5y_3 + x_5y_6 + x_5y_7 + x_7y_5 \\ + x_2 + x_3 + x_4 + y_0 + y_6 + 1,$$

$$f_{13} = x_0y_4 + x_0y_6 + x_1y_5 + x_2y_1 + x_4y_0 + x_5y_0 \\ + x_5y_1 + x_5y_3 + x_5y_6 + x_6y_6 + x_7y_1 + x_7y_2 \\ + x_7y_3 + x_2 + y_5.$$

where x_i is the Sbox input bits, and y_i is the Sbox output bits.

Table 2 presents the number of additional binary variables required to transform the equations system of one Sbox and the nonlinear layer of the AES cipher to the QUBO problem for three different systems defining the Sbox. Finally, the obtained equations system allows for reducing the number of

additional binary variables for the non-linear layer by 60% in relation to the overdefined system defining the Sbox, and by 1% with regard to the system from our previous paper.

The equation systems describing the entire AES cipher for three polynomial equation systems defining the Sbox were transformed into the QUBO problem. The obtained sizes of the QUBO problem are presented in Tab. 3. The use of the efficient system found allows to reduce the target QUBO problem by approximately 58% compared to the use of the overdefined system, and by 1.7% for AES-128 and 2.5% for AES-192 and AES-256 compared to the use of the system of equations from our previous paper.

4. Conclusions

In this paper, we presented a method of searching for an equations system that unambiguously defines the substitution box of the AES cipher and results in the smallest number of variables in the target QUBO problem currently known. As part of the presented research, we defined an optimization problem in which the objective function determining the efficient system was described in the context of its transformation to the QUBO problem, and we proposed specific methods to solve this problem. Contrary to the classical tools used for algebraic attacks, we do not need to construct overdefined equation systems for the proposed method. Therefore, to the best of our knowledge, this is the first analysis of the substitution box of the AES cipher aimed at finding its minimum system of equations.


It is also worth adding that the efficient system identified allows to reduce the size of the target QUBO problem by almost 500 variables for AES-128, compared to our previous result [4]. Assuming the time complexity of solving the QUBO problem equals $O(e^{\sqrt{N}})$, where N is the number of logical variables, it results in a four-time decrease in the amount of time required to perform that algebraic attack on AES-128 when using quantum annealing.

References

- [1] P. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", *IEEE Proceedings 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, USA, 1994 (<https://doi.org/10.1109/SFCS.1994.365700>).
- [2] C.H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and Weaknesses of Quantum Computing", *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1510–1523, 1997 (<https://doi.org/10.1137/S0097539796300933>).
- [3] L.K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search", *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219, 1996 (<https://doi.org/10.1145/237814.237866>).
- [4] E. Burek, M. Wroński, K. Mank, and M. Misztal, "Algebraic Attacks on Block Ciphers Using Quantum Annealing", *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 678–689, 2022 (<https://doi.org/10.1109/TETC.2022.3143152>).


- [5] I.G. Rosenberg, "Reduction of Bivalent Maximization to the Quadratic Case", *Cahiers du Centre d'Etudes de Recherche Operationnelle*, vol. 17, pp. 71–74, 1975.
- [6] N.T. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", *Lecture Notes in Computer Science*, vol. 2501, 2002 (https://doi.org/10.1007/3-540-36178-2_17).
- [7] A.I. Pakhomchik, V.V. Voloshinov, V.M. Vinokur, and G.B. Lesovik, "Converting of Boolean Expression to Linear Equations, Inequalities and QUBO Penalties for Cryptanalysis", *Algorithms*, vol. 15, no. 2, art. no. 33, 2022 (<https://doi.org/0.3390/a15020033>).

Elżbieta Burek, Ph.D.


Assistant Professor
Institute of Mathematics and Cryptology
Faculty of Cybernetics
 <https://orcid.org/0000-0003-2937-0833>
E-mail: elzbieta.burek@wat.edu.pl

Military University of Technology, Warsaw, Poland
<https://www.wat.edu.pl>

Krzysztof Mańk, M.Sc.

Assistant Professor
Institute of Mathematics and Cryptology
Faculty of Cybernetics
 <https://orcid.org/0000-0002-5048-9049>
E-mail: krzysztof.mank@wat.edu.pl
Military University of Technology, Warsaw, Poland
<https://www.wat.edu.pl>

Michał Wroński, Ph.D.

Cybersecurity Expert
Department of Cryptology
 <https://orcid.org/0000-0002-8679-9399>
E-mail: michal.wronski@nask.pl
NASK National Research Institute, Warsaw, Poland
<https://www.nask.pl>