IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 27 October 2021; revised 10 January 2022; accepted 11 January 2022.
Date of publication 21 January 2022; date of current version 7 June 2022.

Digital Object Identifier 10.1109/TETC.2022.3143152

Algebraic Attacks on Block Ciphers Using Quantum Annealing

ELŻBIETA BUREK, MICHAŁ WROŃSKI, KRZYSZTOF MAŃK, AND MICHAŁ MISZTAL

The authors are with the Department of Cybernetics, Military University of Technology, 00-908 Warsaw, Poland
CORRESPONDING AUTHOR: MICHAŁ WROŃSKI (michal.wronski@wat.edu.pl)

This work was supported by 10.13039/501100006141-Wojskowa Akademia Techniczna under Grant 858/2021.

ABSTRACT This paper presents the transformation method of the system of algebraic equations describing the symmetric cipher into the QUBO problem. After transformation of given equations $f_0, f_1, \ldots, f_{n-1}$ to equations over integers $f'_0, f'_1, \ldots, f'_{n-1}$, one can linearize each, obtaining $f'_{lin_i} = lin(f'_i)$, for $i = \overline{0, n-1}$, where *lin* denotes linearization operation. Finally, one can obtain problem in the QUBO form as $(f'_{lin_0})^2 + \cdots + (f'_{lin_{n-1}})^2 + Pen - C$, where *Pen* denotes penalties obtained during linearization of equations, n is the number of equations and C is constant appearing in the polynomial $(f'_{lin_0})^2 + \cdots + (f'_{lin_{n-1}})^2 + Pen$. This paper presents the transformation method of SPN block ciphers to the QUBO problem. What is more, we present the results of the transformation of the complete AES-128 cipher to the QUBO problem, where the number of variables of the equivalent QUBO problem equals approximately 30,026. It is worth noting that AES-128 is much easier to solve using quantum annealing than the factorization problem and the discrete logarithm problem of a similar level of security. For example, factorizing a 3072 bit long RSA integer using quantum annealing requires a QUBO problem of about 2,360,000 variables.

INDEX TERMS Cryptanalysis, AES, block ciphers, algebraic attacks, quantum annealing

I. INTRODUCTION

Quantum computing is one of the most promising approaches that may be used for the cryptanalysis of many cryptographic algorithms. The first major paper in this field was published by Peter W. Shor [1], where he described a quantum algorithm for factorization and discrete logarithm computation. Since that time, many efforts have been made to construct quantum computers, which would break algorithms used in real-world applications. However, the progress in this field is huge, and till now, the biggest quantum computer made by Google has 72 working qubits [2], it is still too little to break real-world algorithms.

In the last few years, the second quantum computing approach has gained much popularity. This approach is quantum annealing, and it is applied in computers built by D-Wave company. Unfortunately, such computers may solve only Ising problems. Of course, other problems as QUBO (Quadratic Unconstrained Binary Optimization) and DQM (Discrete Quadratic Model) may be transformed to the Ising problem.

The quantum annealing approach allowed for some success in factorization, where the quantum factorization record had belonged to the D-Wave computer for some time. Using transformation of integer factorization to the QUBO problem, Dridi and Alghassi [3] were able to factorize integer 200,099, which result was later beaten by Jiang *et al.* [4], and by Wang *et al.* [5], who factorized 20-bit integer 1,028,171. Such transformation of factorization problem to the QUBO problem requires approximately $\frac{n^2}{4}$ logical qubits. It means that factorizing a 3072 bit long RSA integer using quantum annealing requires a QUBO problem of about 2,360,000 variables. It is believed that such an RSA problem has (nowadays) a similar level of security (128 bits) as AES-128.

Moreover, it is also possible to transform discrete logarithm problem to the QUBO problem [6], where approximately $2n^2$ logical qubits are necessary. It means that computing discrete logarithm modulo 3072-bit long prime using quantum annealing requires a QUBO problem of about 18,875,000 variables. It is believed that such a discrete logarithm problem has (nowadays) a similar level of security (128 bits) as AES-128. These arguments make that quantum annealing may be used in cryptanalysis of cryptographic algorithms.

This paper presents a method for transforming algebraic equations of a symmetric cipher into the QUBO problem.

After such transformation, obtained QUBO problem may be solved using the quantum annealing approach, especially on D-Wave computers. At first, algebraic equations of cipher have to be obtained. The idea here is the same as in the case of algebraic attacks. After obtaining Boolean equations of given cipher in algebraic normal form, each equation f has to be transformed to the equation of Boolean variables with integer coefficients as $f' = f - 2k$, where k is an integer, $k \leq \lfloor \frac{f_{max}}{2} \rfloor$ and f_{max} is maximal value polynomial f can take. Moreover, k has to be written as the sum of Boolean variables

$$k = \sum_{i=0}^{bl(f_{max})-1} 2^i k_i + f_{max} + 1 - 2^{bl(f_{max}-1)}, \tag{1}$$

where $bl(x)$ denotes bit-length of integer x. After transformation of given equations, one has to linearize each, obtaining $f'_{lin_i} = lin(f'_i)$, where lin denotes linearization operation. Finally, one can obtain problem in the QUBO form as $(f'_{lin_0})^2 + \cdots + (f'_{lin_{n-1}})^2 + Pen - C$, where Pen denotes penalties obtained during linearization of equations, n is the number of equations and C is constant appearing in the polynomial $(f'_{lin_0})^2 + \cdots + (f'_{lin_{n-1}})^2 + Pen$.

In this paper, we present the results of the transformation of the complete AES-128 cipher to the QUBO problem, where the number of variables of equivalent QUBO problem is equal to approximately 30,026 which means that, at least theoretically, that problem may be solved using the D-Wave Advantage computer. Unfortunately, it is hard to estimate the time this process would require.

II. THE IDEA OF ALGEBRAIC ATTACKS

Algebraic attacks [7], [8], [9], [10] (rather than statistical like differential and linear cryptanalysis) in nature exploit the internal algebraic structure of the algorithm. The general idea is quite simple. First, obtain a representation of the cipher as a system of equations. And then try to solve it to recover unknowns which are secret key bits. In theory, most modern (block and stream) ciphers can be described by a system of multivariate polynomials over a finite field. For the majority of the ciphers, such systems are too complex for any practical solving method.

The term "Algebraic Attack" typically refers to the technique of expressing the whole cryptosystem as a large system of multivariate polynomial equations. To obtain a polynomial system from block ciphers, we must include linear equations from the diffusion layer, key addition, nonlinear equations from the substitution layer, and Key Schedule equations. For the nonlinear equations, we distinguish two cases: explicit equations, which are equations of the form $y_i = g_i(x_0, x_1, \ldots, x_{n-1})$, and implicit equations which are equations of the form $f(x_0, \ldots, x_{n-1}; y_0, \ldots, y_{m-1}) = 0$. We usually consider algebraic attacks when these equations have a small degree.

When mounting an algebraic attack for each nonlinear component of the cipher, we usually attempt to obtain as many low-degree, linearly independent equations as possible. That over-defined systems are generally easier to solve.

In its general form, an algebraic attack is mounted by expressing the whole cipher operation as a system of low-degree multivariate equations, involving the (known) plaintext and ciphertext values, the secret key, and many intermediate variables arising in the cipher operations. It results in huge systems (typically over $GF(2)$). Attack usually requires only one single plaintext/ciphertext pair. The solution of the system is equivalent to key recovery.

It means that for algebraic attacks, we need efficient algorithms for solving algebraic systems. So, the methods for solving polynomials systems are the essential ingredients of algebraic attacks and have recently started receiving special attention from the cryptographic community. The most common methods used in cryptology may include the Linearisation principle, XL and variants, Groebner Basis algorithms (e.g., Buchberger, F4, F5), SAT-solvers, and others.

Linearization is a well-known technique for solving large systems of multivariate polynomial equations. We consider all monomials in the system as independent variables and solve the system using linear algebra techniques (i.e., Gaussian reduction). The method's effectiveness depends on the number of linearly independent polynomials in the system. In the case of Boolean functions with n variables, the total number of monomials of degree d is $N = \sum_{i=1}^{d} \binom{n}{i}$. The complexity of Gaussian reduction is $O(N^3)$. We may theoretically write $O(N\omega)$, where $\omega \approx 2 + \epsilon$, if the matrix of the linearized system is sparse. Note that the problem of estimating the rank of the linearized system is challenging. In order to apply the linearization method, the number of linear equations in the system needs to be approximately the same as the number of monomials in the system. Many techniques have been proposed to generate enough linear equations when this is not the case.

III. FROM ALGEBRAIC ATTACK TO QUBO PROBLEM

This section will present the transformation from the MQ problem to the QUBO problem for block ciphers of the SPN type.

A. GENERATE THE SYSTEM OF MULTIVARIATE QUADRATIC EQUATIONS

Most block ciphers can be described as a system of multivariate equations over $GF(2)$. If in such a system all equations are quadratic, then when looking for a solution we refer to the MQ problem (Multivariate Quadratic Problem). In [11], [12] Shamir *et al.* showed that the MQ problem is NP-hard, and its complexity decreases when the MQ system is over-defined. In [8] Courtois and Pieprzyk showed how to create over-defined systems for Rijndael and Serpent ciphers.

Our approach of creating systems of multivariable quadratic equations to describe a cipher is slightly different. We are looking for a minimal system that still describes the cipher. Minimal in this case means the smallest possible number of equations, variables, and different monomials of degree two because we aim to obtain the smallest possible number of binary variables in the target form of the QUBO problem.

Additional unknown variables were introduced to describe the SPN block cipher encryption algorithm using quadratic

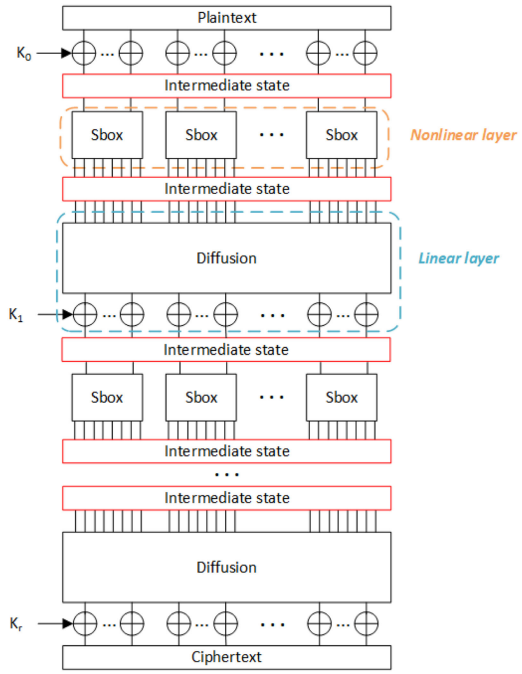

FIGURE 1. Split a round of an SPN block cipher by using additional variables.

equations to describe intermediate states between rounds. Additionally, successive variables were used to represent the intermediate states in each round by the layer of substitution boxes. Such a division of the cipher round, the separation of the linear layer from the nonlinear layer, causes the number of monomials in the system to be smaller than when considering the round as a whole. Figure 1 shows the proposed division of the SPN block cipher round, where the states described by additional unknowns are shown in red.

The unknown in the generated system of multivariate quadratic equations is round key bits and intermediate state bits. Since exactly one solution is sought, the independent variables of the round keys must be related using the equations describing the algorithm for generating the round keys.

The key expansion algorithm is partitioned using additional unknown variables, separating the linear and nonlinear layers like the encryption algorithm.

1) LINEAR LAYER

In the partition proposed by us, the linear layer consists of adding the round key and linear diffusion. Adding a round key is a xor operation of each bit, while the diffusion layer can be a superposition of any linear operations. Hence, the system of equations describing the linear layer of each round consists of as many linear equations as there are bits output from the linear layer.

2) NONLINEAR LAYER

It was assumed that the nonlinear layer consists of some substitution boxes, the so-called S-boxes, with the same number of input and output bits. We assume that the S-box is a $GF(2)^n \rightarrow GF(2)^n$. Therefore, the method of generating implicit multivariate equations of a degree of at most two for over-defined systems, presented in [8], was used. Bijection, where n is the number of input and output bits. Each output bit's equation in algebraic normal form should have a high algebraic degree for a properly designed substitution box. This method allows one to construct equations containing only monomials from a predefined set of acceptable monomials.

For a substitution box of size n, the number of all possible entries into the box is 2^n, the number of all possible monomials is 2^{2n}, the number of all possible monomials of degree two is $\binom{2n}{2}$, and the number of all possible monomials of degree two at most is equal to $\binom{2n}{2} + 2n + 1$. A matrix of dimension $\binom{2n}{2} + 2n + 1 \times 2^n$ was considered, where each row contains the values of a given monomial for all possible inputs. Then, Gaussian elimination was performed while simultaneously storing the operations performed. At least $\binom{2n}{2} + 2n + 1 - 2^n$ rows have been cleared. Hence there are at least $\binom{2n}{2} + 2n + 1 - 2^n$ quadratic equations satisfying a substitution box. For the bigger S-boxes (for example, $n = 8$), the number of rows may be less than the number of columns, which means that quadratic equations satisfying a substitution box may not exist (the value of $\binom{2n}{2} + 2n + 1 - 2^n$ is negative).

Unfortunately, it is a method that generates an over-defined system. Therefore the obtained set of quadratic equations was searched to find the minimum system. For each possible combination of quadratic equations from the obtained set, it was checked whether a given system is satisfied for only one output and whether all outputs were satisfied for each possible input. It means that only given (the only one) S-box satisfies the minimum system of equations. As will be presented later for Simplified AES, we may obtain the minimal system consisting of 5 equations, and in the case of AES-128, we may obtain the minimal system consisting of 12 equations. However, as we will clarify, the better is a system consisting of 13 equations for our purpose.

The selected system should meet the following conditions:

- the minimum number of different square monomials in the system,
- the minimum number of equations in the system.

Since the number of all monomials in the system also affects the number of binary variables in the target form of the QUBO problem, the next step was to check whether it is possible to replace the given equation in the system with another equation with a smaller number of monomials, obtained from the operation xor on two, three, etc. equations.

The system of quadratic equations of many variables obtained as a result of the search was used to describe the substitution box of the nonlinear layer.

Let r be the number of quadratic equations in the found system and m be the number of substitution boxes in the nonlinear layer. Then the entire nonlinear layer of a single round can be described by $r \cdot m$ multivariate quadratic equations.

B. TRANSFORMATION TO PSEUDO-BOOLEAN FUNCTIONS

Pseudo-Boolean functions have been known since 1968 when Hammer in [13] introduced the following definition: Let R be the field of the real numbers; by a pseudo-Boolean function, we shall mean a function

$$f : B_2^n \to R.$$

In a special case, the pseudo-Boolean function is a mapping of B_2^n to a ring Z of integers. If we assume that elements 0 and 1 of B_2 are equated with the numbers 0 and 1, then each Boolean function is also a pseudo-Boolean function.

The following theorem [13] is also well known:

Every pseudo-Boolean function may be written as a polynomial, which is linear in each variable, and which, after the reduction of the similar terms, is uniquely determined up to the order of sums and products.

C. LINEARIZATION

It is crucial that one can obtain the QUBO problem in two little different ways:

1) one can at first linearize each equation f_i' to obtain the linearized equation f'_{lin_i}, then compute the sum $F_{Pen} = \sum_{i=0}^{n-1} (f'_{lin_i})^2 + Pen - C$, where Pen denotes penalties obtained during linearization, n is the number of equations and C is constant appearing in the polynomial $\sum_{i=0}^{n-1} (f'_{lin_i})^2 + Pen$; polynomial F_{Pen} is in such a case in QUBO form;
2) one can at first compute the sum $F = \sum_{i=0}^{n-1} (f_i')^2$, and then make of quadratization of the polynomial F, obtaining F_{Quadr}, finally obtaining polynomial $F_{Pen} = F_{Quadr} + Pen - C$ in QUBO form, where Pen denotes penalties obtained during linearization, n is the number of equations and C is constant appearing in the polynomial $\sum_{i=0}^{n-1} (f'_{lin_i})^2 + Pen$; polynomial F_{Pen} is in such a case in QUBO form.

In the proposed transformation of the system of multivariate polynomial equations to the QUBO problem, the first method simply allows us to compute the maximal number of required variables in the resulting QUBO problem. Therefore, the first method was used.

Linearizing Boolean functions is NP-hard, but we do not need an optimal solution.

The reduction by substitution algorithm proposed by Rosenberg in [14] was used for our research. The reduction is to replace each quadratic monomial with a new variable. Since the QUBO problem is unconstrained, add a penalty to enforce equality after substitution. The reduction is performed as follows:

$$x_i x_j \to x_k + 2(x_i x_j - 2x_k(x_i + x_j) + 3x_k), \tag{2}$$

where $2(x_i x_j - 2x_k(x_i + x_j) + 3x_k)$ is the penalty.

If $x_i x_j = x_k$ then the penalty is zero. Otherwise if $x_i x_j \neq x_k$ then

$$x_i x_j \; < \; x_k + 2(x_i x_j - 2x_k(x_i + x_j) + 3x_k), \tag{3}$$

where the penalty is of high value and such a solution should be rejected when looking for the minimum function.

The cost of this reduction is one new variable per substitution, but a new variable can reduce a given quadratic monomial in many system equations. The biggest problem is determining the optimal order of substitutions. In our research, substitutions were performed according to the order of occurrence of successive square monomials.

The determined penalty after all substitutions is multiplied by a large positive constant and added to the cost function of the generated QUBO problem.

D. TRANSFORMATION OF A LINEAR SYSTEM TO QUBO PROBLEM

In [15], [16], [17], and [18], the use of quantum annealing for the linear least squares problem is presented. Let t denote the number of multivariable polynomial equations in the system describing the cipher, and let p denote the number of binary variables in this system after linearization. Let us define the system of Boolean equations

$$\begin{cases} f_0(x_0, x_1, \ldots, x_{p-1}) \equiv 0 \ (mod\ 2), \\ f_1(x_0, x_1, \ldots, x_{p-1}) \equiv 0 \ (mod\ 2), \\ \vdots \\ f_{t-1}(x_0, x_1, \ldots, x_{p-1}) \equiv 0 \ (mod\ 2). \end{cases} \tag{4}$$

This system may be transformed into the system of Boolean variables with integer coefficients:

$$\begin{cases} f_0(x_0, x_1, \ldots, x_{p-1}) = 2k_0 \\ f_1(x_0, x_1, \ldots, x_{p-1}) = 2k_1 \\ \vdots \\ f_{t-1}(x_0, x_1, \ldots, x_{p-1}) = 2k_{t-1}. \end{cases} \tag{5}$$

Since the equations are pseudo-Boolean functions, each equation is equal to zero modulo 2, which means it is equal to a multiple of 2.

To determine the value of $2k_i$, we assume the maximum possible value of the given equation, i.e., we assume that all binary variables of the equation are equal to 1, which is equivalent to the number of monomials in the equation. The determined maximum value of k_i determines the number of binary variables by means of which the actual value of the equation f_i will be presented. Hence, each k_i is a polynomial of degree 1 at most, for which we use successive binary variables.

Now let s denote the number of binary variables used for the construction of all of the polynomials f_i and k_i.

Let us present the resulting system of residuals:

$$\begin{cases} f_0(x_0, x_1, \ldots, x_{p-1}) - 2k_0 = 0 \\ f_1(x_0, x_1, \ldots, x_{p-1}) - 2k_1 = 0 \\ \vdots \\ f_{t-1}(x_0, x_1, \ldots, x_{p-1}) - 2k_{t-1} = 0 \end{cases} \tag{6}$$

in matrix form:

$$\mathbf{A}x + \mathbf{c} = 0 \tag{7}$$

where:

- $\mathbf{A}$ is the $t \times s$ matrix of the coefficients of the polynomials f_i and k_i,
- x is a vector of $s \times 1$ of binary variables $x_0, x_1, \ldots, x_{s-1}$, occurring in the polynomials f_i and k_i,
- $\mathbf{c}$ is a vector of $t \times 1$ of values 0 or 1 which are constants of f_i polynomials.

$$\mathbf{A}x + \mathbf{c} = \begin{pmatrix} A_{0,0} & A_{0,1} & \ldots & A_{0,s-1} \\ A_{1,0} & A_{1,1} & \ldots & A_{1,s-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{t-1,0} & A_{t-1,1} & \ldots & A_{t-1,s-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{s-1} \end{pmatrix} + + \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{t-1} \end{pmatrix}. \tag{8}$$

This is equivalent to

$$\mathbf{A}x + \mathbf{c} = \begin{pmatrix} A_{0,0}x_0 + A_{0,1}x_1 + \cdots + A_{0,s-1}x_{s-1} + c_0 \\ A_{1,0}x_0 + A_{1,1}x_1 + \cdots + A_{1,s-1}x_{s-1} + c_1 \\ \vdots \\ A_{t-1,0}x_0 + A_{t-1,1}x_1 + \cdots + A_{t-1,s-1}x_{s-1} + c_{t-1} \end{pmatrix}. \tag{9}$$

The 2nd square norm of the vector of residuals in Equation 9 is determined as:

$$\|\mathbf{A}x + \mathbf{c}\|_2^2 = \sum_{i=0}^{t-1} \left(|A_{i,0}x_0 + A_{i,1}x_1 + \cdots + A_{i,s-1}x_{s-1} + c_i|\right)^2 \tag{10}$$

$$\begin{aligned} &\|\mathbf{A}x + \mathbf{c}\|_2^2 = \\ &= \sum_{i=0}^{t-1} (A_{i,0}^2 x_0^2 + 2A_{i,0}c_0x_0 + A_{i,1}^2 x_1^2 + 2A_{i,1}c_1x_1 + \cdots + \\ &\quad + A_{i,s-1}^2 x_{s-1}^2 + 2A_{i,s-1}c_{s-1}x_{s-1}) + \\ &\quad + \sum_{i=0}^{t-1} (2A_{i,0}A_{i,1}x_0x_1 + 2A_{i,0}A_{i,2}x_0x_2 + \cdots + \\ &\quad + 2A_{i,s-2}A_{i,s-1}x_{s-2}x_{s-1}) + \\ &\quad + \sum_{i=0}^{t-1} c_i^2 \end{aligned} \tag{11}$$

$$\begin{aligned} &\|\mathbf{A}x + \mathbf{c}\|_2^2 = \\ &\sum_{i=0}^{t-1} \sum_{j=0}^{s-1} A_{i,j}(A_{i,j} + 2c_i)x_j + \sum_{i=0}^{t-1} 2 \sum_{j<k} A_{i,j}A_{i,k}x_jx_k + \sum_{i=0}^{t-1} c_i^2 = \\ &= \sum_j B_{j,j}x_j + \sum_{j<k} B_{j,k}x_jx_k + C = x^T\mathbf{B}x + C \end{aligned} \tag{12}$$

where

$$B_{j,j} = \sum_{i=0}^{t-1} A_{i,j}(A_{i,j} + 2c_i), \tag{13}$$

$$B_{j,k} = 2\sum_{i=0}^{t-1} A_{i,j}A_{i,k}, \tag{14}$$

$$C = \sum_{i=0}^{t-1} c_i^2, \tag{15}$$

and for every binary variable x holds that $x^2 = x$.

Finding the values of binary variables, including key variables, solves the following constraint problem

$$\min_{x \in \{0,1\}^s} y = x^T\mathbf{B}x + C, \tag{16}$$

where the constraints are the substitutions of new binary variables during linearization.

The square penalty determined during linearization, corresponding to the constraints of our problem, is presented in the form

$$x^T\mathbf{D}x. \tag{17}$$

The optimal solution to the constrained problem is the same as the unconstrained problem with the penalty. Hence our problem is as follows:

$$\min_{x \in \{0,1\}^s} y = x^T\mathbf{B}x + x^T\mathbf{D}x + C = x^T\mathbf{Q}x + C. \tag{18}$$

Since x_i are binary variables, we obtain problem in the QUBO form

$$\min_{x \in \{0,1\}^s} y = x^T\mathbf{Q}x. \tag{19}$$

The constant C is not included in the QUBO form.

Summing up, variables k_i's are introduced while Boolean equations are transformed into equations of Boolean variables, but with integer coefficients. Even though k_i's are unknown until the system is solved. During linearization, auxiliary variables are introduced, and knowledge about basic variables would be lost if penalties would not be introduced. If penalties were not introduced, then it would be very probable that the solution of the linear system we obtained would not satisfy the substitutions. To ensure that substitutions will be made correctly, we have to keep all the penalties to 0.

To clarify our approach, we will show how to solve a very simple system of equations of Boolean variables x_0, x_1, x_2

$$\begin{cases} x_0x_1 + x_2 + 1 \equiv 0 \ (mod\ 2), \\ x_1x_2 + x_0 \equiv 0 \ (mod\ 2), \\ x_0 + x_1 + x_2 + 1 \equiv 0 \ (mod\ 2). \end{cases} \tag{20}$$

This system may be transformed into the system of equations of Boolean variables with integer coefficients

$$\begin{cases} f_0 : x_0x_1 + x_2 + 1 = 2k_0, \\ f_1 : x_1x_2 + x_0 = 2k_1, \\ f_2 : x_0 + x_1 + x_2 + 1 = 2k_2, \end{cases} \tag{21}$$

which is equivalent to

$$\begin{cases} f_0' : x_0x_1 + x_2 + 1 - 2k_0 = 0, \\ f_1' : x_1x_2 + x_0 - 2k_1 = 0, \\ f_2' : x_0 + x_1 + x_2 + 1 - 2k_2 = 0, \end{cases} \tag{22}$$

At first, the system (22) will be linearized. We will make the following substitutions $x_3 = x_0x_1$ and $x_4 = x_1x_2$ for which we will obtain the following penalties: $Pen_1 = 2(x_0x_1 - 2x_3(x_0 + x_1) + 3x_3)$ and $Pen_2 = 2(x_1x_2 - 2x_4(x_1 + x_2) + 3x_4)$. The linearized system of equations is presented by the system (23)

$$\begin{cases} f_{lin_0}' : x_3 + x_2 + 1 - 2k_0 = 0, \\ f_{lin_1}' : x_4 + x_0 - 2k_1 = 0, \\ f_{lin_2}' : x_0 + x_1 + x_2 + 1 - 2k_2 = 0. \end{cases} \tag{23}$$

The values of k_0, k_1, k_2 are limited by the maximal value of each of the equations. Because the maximal value of $x_3 + x_2 + 1$ is 3, it must hold that $2k_0 \leq 3$ and the maximal value of k_0 is 1. Similarly, because the maximal value of $x_4 + x_0$ is 2, it must hold that $2k_1 \leq 2$ and the maximal value of k_1 is also 1. Finally, because the maximal value of $x_0 + x_1 + x_2 + 1$ is 4, it must hold that $2k_2 \leq 4$ and the maximal value of k_2 is 2. k_0, k_1, k_2 need to be presented as the sum of successive Boolean variables with integer coefficients (see Equation (1)). It means that $k_0 = x_5, k_1 = x_6, k_2 = x_7 + x_8$. The system (23) may be presented as (24)

$$\begin{cases} x_3 + x_2 + 1 - 2x_5 = 0, \\ x_4 + x_0 - 2x_6 = 0, \\ x_0 + x_1 + x_2 + 1 - 2(x_7 + x_8) = 0. \end{cases} \tag{24}$$

Finally, one can compute $F_{Pen} = (f_{lin_0}')^2 + (f_{lin_1}')^2 + (f_{lin_2}')^2 + c \cdot Pen - C$, where $Pen = Pen_1 + Pen_2$ denotes penalties obtained during linearization and $c = 10$ is a constant (in here $c = 10$ is chosen arbitrarily; in general a constant should be so large to ensure that for improper values of variables it should be impossible to obtain minimal energy), and C is constant appearing in the polynomial $(f_{lin_0}')^2 + (f_{lin_1}')^2 + (f_{lin_2}')^2 + c \cdot Pen$, which Polynomial has the following form $F_{Pen}' = 22x_0x_1 + 2x_0x_2 + 22x_1x_2 - 40x_0x_3 - 40x_1x_3 + 2x_2x_3 + 2x_0x_4 - 40x_1x_4 - 40x_2x_4 - 4x_2x_5 - 4x_3x_5 - 4x_0x_6 - 4x_4x_6 - 4x_0x_7 - 4x_1x_7 - 4x_2x_7 - 4x_0x_8 - 4x_1x_8 - 4x_2x_8 + 8x_7x_8 + 4x_0 + 3x_1 + 6x_2 + 63x_3 + 61x_4 + 4x_6 + 2$.

Finally, $F_{Pen} = F_{Pen}' - 2$, because 2 is constant appearing in the polynomial F_{Pen}'.

F_{Pen} is of course in the QUBO form. The minimal energy for this problem (minimal energy is equal to -2; the minimal energy is the negation of constant C) may be obtained for the proper solution. There are two proper solutions. The first one is $x_0 = 0, x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0, x_5 = 1, \quad x_6 = 0, x_7 = 1, x_8 = 0$ and the second one is $x_0 = 0, x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0, x_5 = 1, x_6 = 0, x_7 = 0, x_8 = 1$. However, it is important to see that these both solutions give the same solution of System (23) because $k_2 = 1$ both for $k_2 = x_7 + x_8$ if $x_7 = 1, x_8 = 0$ and $k_2 = x_7 + x_8$ if $x_7 = 0, x_8 = 1$.

Of course solution $x_0 = 0, x_1 = 0, x_2 = 1$ satisfy the System (20), from which we began.

What is more, one can also use the matrix notation. According to Equation (7) holds that

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & -2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -2 & -2 \end{bmatrix}, \tag{25}$$

and

$$\mathbf{c} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}. \tag{26}$$

According to Equations (18) and (19) matrices $\mathbf{B}, \mathbf{D}, \mathbf{Q}$ are equal to, respectively

$$\mathbf{B} = \begin{bmatrix} 4 & 2 & 2 & 0 & 2 & 0 & -4 & -4 & -4 \\ 0 & 3 & 2 & 0 & 0 & 0 & 0 & -4 & -4 \\ 0 & 0 & 6 & 2 & 0 & -4 & 0 & -4 & -4 \\ 0 & 0 & 0 & 3 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{27}$$

$$\mathbf{D} = \begin{bmatrix} 0 & 20 & 0 & -40 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & -40 & -40 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -40 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 60 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 60 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{28}$$

$$\mathbf{Q} = \begin{bmatrix} 4 & 22 & 2 & -40 & 2 & 0 & -4 & -4 & -4 \\ 0 & 3 & 22 & -40 & -40 & 0 & 0 & -4 & -4 \\ 0 & 0 & 6 & 2 & -40 & -4 & 0 & -4 & -4 \\ 0 & 0 & 0 & 63 & 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 61 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{29}$$

The constant C is equal to -2.

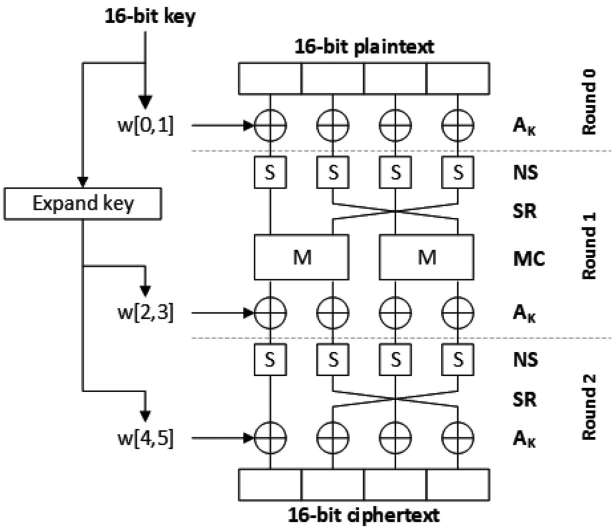

FIGURE 2. Structure of the Simplified AES encryption algorithm. [20].

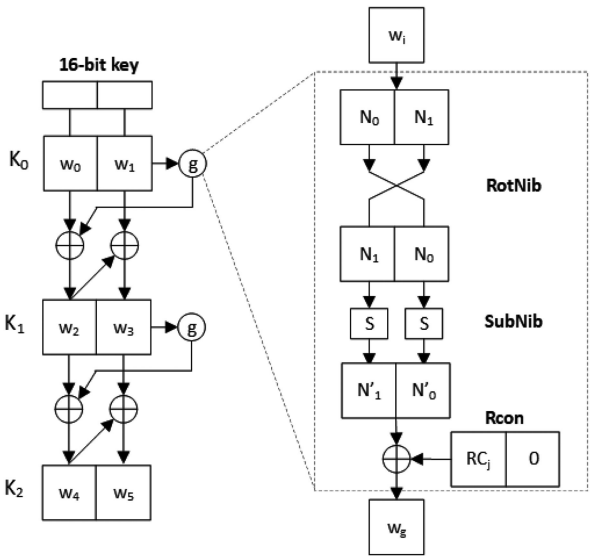

FIGURE 3. Simplified AES key expansion algorithm [20].

IV. APPLICATION OF OUR APPROACH TO THE AES CIPHER

We will explain the transformation of the cipher's multivariable polynomial equations to the QUBO problem on the example of the Simplified AES cipher, which was developed by Edward Schaefer [19].

A. TRANSFORMATION OF THE SYSTEM OF MULTIVARIATE POLYNOMIAL EQUATIONS TO THE QUBO PROBLEM FOR SIMPLIFIED AES CIPHER

The overall structure of Simplified AES is shown in Figure 2. The input of the Simplified AES encryption algorithm is a 16-bit plaintext block and a 16-bit key. Processing a single input block produces a 16-bit ciphertext.

The encryption algorithm is performed in three rounds that contain four different functions: add key (A_K), nibble substitution (*NS*), shift row (*SR*), and mix column (*MC*). Round 0 only consists of the add key function. Round 1 is a full round with all four functions. The last Round 2 consists of three different functions; there is no mix column function. Thus, encryption can be presented as a composition of the following functions

$$A_{K_2} \circ SR \circ NS \circ A_{K_1} \circ MC \circ SR \circ NS \circ A_{K_0}$$

The algorithm for generating the round keys of the Simplified AES algorithm is shown in Figure 3. A 16-bit key is grouped into two 8-bit words, from which the following four words are generated, forming another two round keys. The words w_1, w_3, and w_5 are processed by the g function, consisting of three different functions: *RotNib*, *SubNib*, *Rcon*. The *RotNib* function changes the input nibble places. The *SubNib* function uses the same substitution boxes as the encryption algorithm. The *Rcon* function is a bitwise xor operation of the given word with the corresponding constant.

The intermediate state was presented using additional variables to describe the Simplified AES encryption algorithm as a system of multivariable polynomials. Figure 4 shows the partition of the encryption algorithm using intermediate variables marked in red and the order bits of the input and output of the substitution box. The bits of round keys (from x_0 to x_{47}) and intermediate variables (from x_{96} to x_{159}) are unknown in the created system of polynomials. Additionally, to explain our approach, the plaintext and ciphertext bits that are known during the attack are shown here using the variables x_{64} to x_{95}.

Similarly, additional intermediate variables were introduced to the algorithm of generating round keys, separating the linear and nonlinear layers, as shown in Figure 5.

The nonlinear layer of the Simplified AES encryption algorithm consists of four substitution boxes. Each substitution

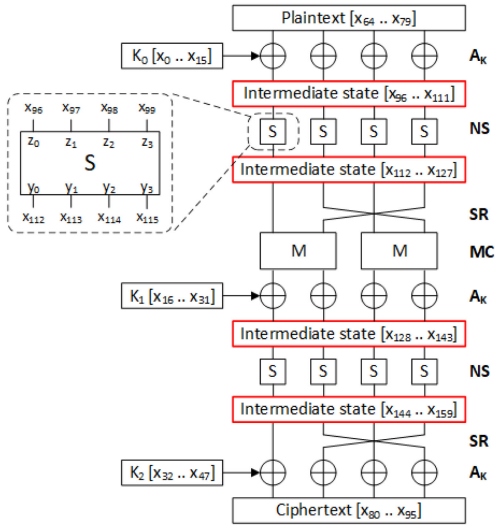

FIGURE 4. Necessary variables for a Simplified AES encryption algorithm.

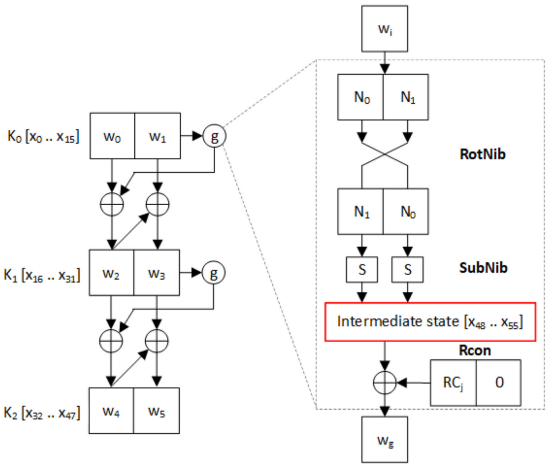

FIGURE 5. Necessary variables for a Simplified AES key expansion algorithm.

box can be described by a system of up to 21 independent quadratic equations that contain 28 different monomials of degree two.

The equations are as follows:

$$eq_0 : z_0z_1 + z_0z_3 + z_0y_2 + z_0 + z_1z_2 + z_1 + y_0 + y_1 + 1 = 0,$$
$$eq_1 : z_0z_1 + z_0z_2 + z_0y_0 + z_0y_1 + z_0y_2 + z_0 + z_1 + z_2 + y_0 + y_1 + y_2 + 1 = 0,$$
$$eq_2 : z_0z_1 + z_0z_2 + z_0z_3 + z_0y_0 + z_0y_1 + z_0y_3 + z_0 + z_1z_3 + z_1y_0 + z_1 = 0,$$
$$eq_3 : z_0z_1 + z_0z_2 + z_0z_3 + \quad z_0y_0 + z_0y_3 + z_1z_3 + z_1y_1 + y_0 + y_1 + 1 = 0,$$
$$eq_4 : z_0z_1 + z_0z_2 + z_0y_2 + z_0y_3 + z_0 + z_1y_2 = 0,$$
$$eq_5 : z_0z_3 + z_0y_1 + z_0y_2 + z_0 + z_1z_3 + z_1y_3 + z_2 + z_3 + y_0 + y_1 + y_3 = 0,$$
$$eq_6 : z_0z_1 + z_0y_0 + z_0y_2 + z_0y_3 + z_0 + z_1z_3 + z_2z_3 + z_2 + y_0 + y_3 = 0,$$
$$eq_7 : z_0z_1 + z_0z_2 + z_0z_3 + z_0y_2 + z_1z_3 + z_2y_0 + z_3 + y_3 + 1 = 0,$$
$$eq_8 : z_0z_3 + z_0y_0 + z_0y_2 + z_1z_3 + z_2y_1 + z_2 + z_3 + y_3 + 1 = 0,$$
$$eq_9 : z_0z_1 + z_0z_2 + z_0y_0 + z_0y_1 + z_0y_3 + z_0 + z_1 + z_2y_2 + z_2 + y_0 + y_1 + 1 = 0,$$
$$eq_{10} : z_0z_3 + z_0y_0 + z_0y_1 + z_0y_3 + z_0 + z_1z_3 + z_2y_3 + z_2 + y_0 + y_3 = 0,$$
$$eq_{11} : z_0z_1 + z_0z_2 + z_0y_1 + z_0y_3 + z_3y_0 + z_3 + y_0 + 1 = 0,$$
$$eq_{12} : z_0y_0 + z_0y_1 + z_0 + z_2 + z_3y_1 + z_3 + y_0 + y_3 = 0,$$
$$eq_{13} : z_0y_0 + z_0 + z_3y_2 + z_3 + y_0 + 1 = 0,$$
$$eq_{14} : z_0z_1 + z_0z_2 + z_0z_3 + z_0y_0 + z_0y_2 + z_1z_3 + z_3y_3 + z_3 + y_0 + 1 = 0,$$
$$eq_{15} : z_0z_2 + z_0z_3 + z_0y_1 + z_0y_3 + z_2 + z_3 + y_0y_1 + y_0 + y_1 + y_3 = 0,$$
$$eq_{16} : z_0z_1 + z_0y_1 + z_0y_2 + z_0y_3 + z_1 + z_2 + y_0y_2 + y_0 + y_1 + 1 = 0,$$
$$eq_{17} : z_0z_1 + z_0z_2 + z_0y_2 + z_0 + z_1z_3 + y_0y_3 + y_3 = 0,$$
$$eq_{18} : z_0z_2 + z_0y_1 + z_0y_2 + z_0y_3 + z_0 + y_1y_2 = 0,$$
$$eq_{19} : z_0z_2 + z_0y_1 + z_0y_2 + z_0y_3 + z_1z_3 + y_0 + y_1y_3 + y_1 + 1 = 0,$$
$$eq_{20} : z_0z_3 + z_0y_0 + z_0y_3 + z_0 + z_3 + y_0 + y_2y_3 + 1 = 0,$$

where the variables z_i are the input bits to the substitution box and y_i are the output bits. Hence, the entire nonlinear layer in the round can be written using a system of 84 polynomials of degree two, containing 112 different monomials of degree two, which must be replaced with new binary variables during linearization.

Since we strive to obtain the smallest possible number of binary variables in the QUBO problem, the system of 21 polynomial equations was searched to find another system with the minimum number of different monomials of degree two, satisfying the substitution box.

Eight systems were found, consisting of 5 equations and 12 different monomials of degree two, satisfying the substitution box. Of these eight, the system selected had the smallest number of all monomials, equal to 23. The selected system consists of the following equations:

$$eq_0 : z_0z_1 + z_0z_3 + z_0y_2 + z_0 + z_1z_2 + z_1 + y_0 + y_1 + 1 = 0,$$
$$eq_1 : z_0z_1 + z_0z_2 + z_0y_0 + z_0y_1 + z_0y_2 + z_0 + z_1 + z_2 + y_0 + y_1 + y_2 + 1 = 0,$$
$$eq_6 : z_0z_1 + z_0y_0 + z_0y_2 + z_0y_3 + z_0 + z_1z_3 + z_2z_3 + z_2 + y_0 + y_3 = 0,$$
$$eq_{11} : z_0z_1 + z_0z_2 + z_0y_1 + z_0y_3 + z_3y_0 + z_3 + y_0 + 1 = 0,$$
$$eq_{12} : z_0y_0 + z_0y_1 + z_0 + z_2 + z_3y_1 + z_3 + y_0 + y_3 = 0.$$

An attempt was made to find a better system by consecutively performing the xor operation on two, three, and four equations, but with such a small number of equations, it did not benefit.

The Simplified AES Round 0 is a linear layer consisting only of the K_0 key addition operation. This layer can be written in 16 linear equations, the form:

$$f_0 : x_0 + x_{64} + x_{96} = 0,$$
$$f_1 : x_1 + x_{65} + x_{97} = 0,$$
$$f_2 : x_2 + x_{66} + x_{98} = 0,$$
$$f_3 : x_3 + x_{67} + x_{99} = 0,$$
$$f_4 : x_8 + x_{72} + x_{104} = 0,$$
$$f_5 : x_9 + x_{73} + x_{105} = 0,$$
$$f_6 : x_{10} + x_{74} + x_{106} = 0,$$
$$f_7 : x_{11} + x_{75} + x_{107} = 0,$$
$$f_8 : x_4 + x_{68} + x_{100} = 0,$$
$$f_9 : x_5 + x_{69} + x_{101} = 0,$$
$$f_{10} : x_6 + x_{70} + x_{102} = 0,$$
$$f_{11} : x_7 + x_{71} + x_{103} = 0,$$
$$f_{12} : x_{12} + x_{76} + x_{108} = 0,$$
$$f_{13} : x_{13} + x_{77} + x_{109} = 0,$$
$$f_{14} : x_{14} + x_{78} + x_{110} = 0,$$
$$f_{15} : x_{15} + x_{79} + x_{111} = 0.$$

Round 1 of the Simplified AES cipher is a full round consisting of a nonlinear and a linear layer.

Since the substitution box has been described with 5 equations, the entire nonlinear layer can be represented by 20 equations of degree two, containing 48 different monomials of degree two. The equations are as follows:

$$f_{16} : x_{96}x_{112} + x_{96}x_{113} + x_{96} + x_{98} + x_{99}x_{113} + x_{99} + x_{112} + x_{115} = 0,$$
$$f_{17} : x_{96}x_{97} + x_{96}x_{98} + x_{96}x_{113} + x_{96}x_{115} + x_{99}x_{112} + x_{99} + x_{112} + 1 = 0,$$
$$f_{18} : x_{96}x_{97} + x_{96}x_{112} + x_{96}x_{114} + x_{96}x_{115} + x_{96} + x_{97}x_{99} + x_{98}x_{99} + x_{98} + x_{112} + x_{115} = 0,$$

$f_{19} : x_{96}x_{97}+x_{96}x_{98}+x_{96}x_{112}+x_{96}x_{113}+x_{96}x_{114}+x_{96}+x_{97}+x_{98}+x_{112}+x_{113}+x_{114}+1=0,$
$f_{20} : x_{96}x_{97}+x_{96}x_{99}+x_{96}x_{114}+x_{96}+x_{97}x_{98}+x_{97}+x_{112}+x_{113}+1=0,$
$f_{21} : x_{100}x_{116}+x_{100}x_{117}+x_{100}+x_{102}+x_{103}x_{117}+x_{103}+x_{116}+x_{119}=0,$
$f_{22} : x_{100}x_{101}+x_{100}x_{102}+x_{100}x_{117}+x_{100}x_{119}+x_{103}x_{116}+x_{103}+x_{116}+1=0,$
$f_{23} : x_{100}x_{101}+x_{100}x_{116}+x_{100}x_{118}+x_{100}x_{119}+x_{100}+x_{101}x_{103}+x_{102}x_{103}+x_{102}+x_{116}+x_{119}=0,$
$f_{24} : x_{100}x_{101}+x_{100}x_{102}+x_{100}x_{116}+x_{100}x_{117}+x_{100}x_{118}+x_{100}+x_{101}+x_{102}+x_{116}+x_{117}+x_{118}+1=0,$
$f_{25} : x_{100}x_{101}+x_{100}x_{103}+x_{100}x_{118}+x_{100}+x_{101}x_{102}+x_{101}+x_{116}+x_{117}+1=0,$
$f_{26} : x_{104}x_{120}+x_{104}x_{121}+x_{104}+x_{106}+x_{107}x_{121}+x_{107}+x_{120}+x_{123}=0,$
$f_{27} : x_{104}x_{105}+x_{104}x_{106}+x_{104}x_{121}+x_{104}x_{123}+x_{107}x_{120}+x_{107}+x_{120}+1=0,$
$f_{28} : x_{104}x_{105}+x_{104}x_{120}+x_{104}x_{122}+x_{104}x_{123}+x_{104}+x_{105}x_{107}+x_{106}x_{107}+x_{106}+x_{120}+x_{123}=0,$
$f_{29} : x_{104}x_{105}+x_{104}x_{106}+x_{104}x_{120}+x_{104}x_{121}+x_{104}x_{122}+x_{104}+x_{105}+x_{106}+x_{120}+x_{121}+x_{122}+1=0,$
$f_{30} : x_{104}x_{105}+x_{104}x_{107}+x_{104}x_{122}+x_{104}+x_{105}x_{106}+x_{105}+x_{120}+x_{121}+1=0,$
$f_{31} : x_{108}x_{124}+x_{108}x_{125}+x_{108}+x_{110}+x_{111}x_{125}+x_{111}+x_{124}+x_{127}=0,$
$f_{32} : x_{108}x_{109}+x_{108}x_{110}+x_{108}x_{125}+x_{108}x_{127}+x_{111}x_{124}+x_{111}+x_{124}+1=0,$
$f_{33} : x_{108}x_{109}+x_{108}x_{124}+x_{108}x_{126}+x_{108}x_{127}+x_{108}+x_{109}x_{111}+x_{110}x_{111}+x_{110}+x_{124}+x_{127}=0,$
$f_{34} : x_{108}x_{109}+x_{108}x_{110}+x_{108}x_{124}+x_{108}x_{125}+x_{108}x_{126}+x_{108}+x_{109}+x_{110}+x_{124}+x_{125}+x_{126}+1=0,$
$f_{35} : x_{108}x_{109}+x_{108}x_{111}+x_{108}x_{126}+x_{108}+x_{109}x_{110}+x_{109}+x_{124}+x_{125}+1=0.$

16 linear equations of the form describe the Round 1 linear layer:

$f_{36} : x_{16}+x_{112}+x_{126}+x_{128}=0,$
$f_{37} : x_{17}+x_{113}+x_{124}+x_{127}+x_{129}=0,$
$f_{38} : x_{18}+x_{114}+x_{124}+x_{125}+x_{130}=0,$
$f_{39} : x_{19}+x_{115}+x_{125}+x_{131}=0,$
$f_{40} : x_{24}+x_{118}+x_{120}+x_{136}=0,$
$f_{41} : x_{25}+x_{116}+x_{119}+x_{121}+x_{137}=0,$
$f_{42} : x_{26}+x_{116}+x_{117}+x_{122}+x_{138}=0,$
$f_{43} : x_{27}+x_{117}+x_{123}+x_{139}=0,$
$f_{44} : x_{20}+x_{114}+x_{124}+x_{132}=0,$
$f_{45} : x_{21}+x_{112}+x_{115}+x_{125}+x_{133}=0,$
$f_{46} : x_{22}+x_{112}+x_{113}+x_{126}+x_{134}=0,$
$f_{47} : x_{23}+x_{113}+x_{127}+x_{135}=0,$
$f_{48} : x_{28}+x_{116}+x_{122}+x_{140}=0,$
$f_{49} : x_{29}+x_{117}+x_{120}+x_{123}+x_{141}=0,$
$f_{50} : x_{30}+x_{118}+x_{120}+x_{121}+x_{142}=0,$
$f_{51} : x_{31}+x_{119}+x_{121}+x_{143}=0.$

The last round of the Simplified AES cipher, Round 2, similarly to Round 1, consists of a nonlinear and a linear layer. The nonlinear layer, as before, was described by 20 quadratic equations with 48 different quadratic monomials. The form of the equations is analogous to those for Round 1, only the indexes of the variables will change. 16 equations of degree one describe the same for the linear layer.

The algorithm of generating the round keys of the Simplified AES cipher has a nonlinear layer consisting of two substitution boxes. Thus, the equations for word w_2 and analogously for word w_4 were generated in two parts. The first part consists of 10 quadratic equations describing the *RotNib* and *SubNib* operations, while the second part consists of 8 linear equations describing the *Rcon* operation and xor operations with the word w_0. The equations between w_0 and w_2 are:

$f_{88} : x_{12}x_{48}+x_{12}x_{49}+x_{12}+x_{14}+x_{15}x_{49}+x_{15}+x_{48}+x_{51}=0,$
$f_{89} : x_{12}x_{13}+x_{12}x_{14}+x_{12}x_{49}+x_{12}x_{51}+x_{15}x_{48}+x_{15}+x_{48}+1=0,$
$f_{90} : x_{12}x_{13}+x_{12}x_{48}+x_{12}x_{50}+x_{12}x_{51}+x_{12}+x_{13}x_{15}+x_{14}x_{15}+x_{14}+x_{48}+x_{51}=0,$
$f_{91} : x_{12}x_{13}+x_{12}x_{14}+x_{12}x_{48}+x_{12}x_{49}+x_{12}x_{50}+x_{12}+x_{13}+x_{14}+x_{48}+x_{49}+x_{50}+1=0,$
$f_{92} : x_{12}x_{13}+x_{12}x_{15}+x_{12}x_{50}+x_{12}+x_{13}x_{14}+x_{13}+x_{48}+x_{49}+1=0,$
$f_{93} : x_{8}x_{52}+x_{8}x_{53}+x_{8}+x_{10}+x_{11}x_{53}+x_{11}+x_{52}+x_{55}=0,$
$f_{94} : x_{8}x_{9}+x_{8}x_{10}+x_{8}x_{53}+x_{8}x_{55}+x_{11}x_{52}+x_{11}+x_{52}+1=0,$
$f_{95} : x_{8}x_{9}+x_{8}x_{52}+x_{8}x_{54}+x_{8}x_{55}+x_{8}+x_{9}x_{11}+x_{10}x_{11}+x_{10}+x_{52}+x_{55}=0,$
$f_{96} : x_{8}x_{9}+x_{8}x_{10}+x_{8}x_{52}+x_{8}x_{53}+x_{8}x_{54}+x_{8}+x_{9}+x_{10}+x_{52}+x_{53}+x_{54}+1=0,$
$f_{97} : x_{8}x_{9}+x_{8}x_{11}+x_{8}x_{54}+x_{8}+x_{9}x_{10}+x_{9}+x_{52}+x_{53}+1=0,$
$f_{98} : x_{0}+x_{16}+x_{48}=0,$
$f_{99} : x_{1}+x_{17}+x_{49}=0,$
$f_{100} : x_{2}+x_{18}+x_{50}=0,$
$f_{101} : x_{3}+x_{19}+x_{51}=0,$
$f_{102} : x_{4}+x_{20}+x_{52}=0,$
$f_{103} : x_{5}+x_{21}+x_{53}=0,$
$f_{104} : x_{6}+x_{22}+x_{54}=0,$
$f_{105} : x_{7}+x_{23}+x_{55}=0.$

The word w_3 and analogously w_5 is determined by the linear xor operation, represented by the 8 linear equations for each word. The equations for the word w_3 are:

$f_{106} : x_{8}+x_{16}+x_{24}=0,$
$f_{107} : x_{9}+x_{17}+x_{25}=0,$
$f_{108} : x_{10}+x_{18}+x_{26}=0,$
$f_{109} : x_{11}+x_{19}+x_{27}=0,$
$f_{110} : x_{12}+x_{20}+x_{28}=0,$
$f_{111} : x_{13}+x_{21}+x_{29}=0,$
$f_{112} : x_{14}+x_{22}+x_{30}=0,$
$f_{113} : x_{15}+x_{23}+x_{31}=0.$

Thus, 26 equations of degree two or less are related to the Round 2 keys, containing 24 different monomials of degree two.

Combining all the equations, the multivariate quadratic equation system describing the entire Simplified AES cipher consists of 140 equations containing 160 binary variables, 144 different monomials of degree two, and 831 all monomials.

During linearization, 144 different monomials of degree two are changed into new binary variables (x_{160} to x_{303}).

TABLE 1. Results of Transformation of the System of Multivariate Quadratic Equations Describing the Simplified AES Cipher to the QUBO Problem for the Set of 21 and 5 Equations Describing the Substitution Box.

The number of equations in the system describing the substitution box.	21	5
The number of equations describing the cipher.	332	140
The number of binary variables in the cipher description system.	160	160
The number of variables used during linearization.	336	144
The number of binary variables after linearization.	496	304
The number of binary variables for the value of k_i.	807	279
The number of monomials in the obtained QUBO problem.	11950	4522
The number of binary variables in the obtained QUBO problem.	1303	583

Simultaneously, a penalty was assigned, being the sum of 576 monomials of a degree of at most two.

Then, the number of binary variables was determined for the k_i value of each equation, which depends on the number of monomials in the equation. For example, let us take the following linearized equation:

$$f_{lin_{19}}: x_{96} + x_{97} + x_{98} + x_{112} + x_{113} + x_{114} + x_{160} + x_{161} + x_{163} + x_{164} + x_{169} + 1 \equiv 0(mod 2)$$

which is equivalent to the equation:

$$f_{lin_{19}}: x_{96} + x_{97} + x_{98} + x_{112} + x_{113} + x_{114} + x_{160} + x_{161} + x_{163} + x_{164} + x_{169} + 1 - 2k_{19} = 0$$

The maximum value of the pseudo-Boolean function: $x_{96} + x_{97} + x_{98} + x_{112} + x_{113} + x_{114} + x_{160} + x_{161} + x_{163} + x_{164} + x_{169} + 1$ is 12. Thus, the maximum value k_{19} of this equation is 6, which can be represented with three bits. Hence, for the value of k_{19} we take the next three binary variables, obtaining the following equation:

$$f'_{lin_{19}}: x_{96} + x_{97} + x_{98} + x_{112} + x_{113} + x_{114} + x_{160} + x_{161} + x_{163} + x_{164} + x_{169} + 1 - 2(x_{329} + x_{330} + x_{331}) = 0$$

For each equation of the system describing the Simplified AES cipher, the number of binary variables to represent all values of k_i was determined. The number of binary variables needed is 279, these are variables from x_{304} to x_{582}. After calculating the sum of the squares of the residuals and adding the penalty, a problem in the QUBO form was obtained, consisting of 4522 monomials of degree two or less and 583 binary variables.

The presented transformation was also performed for the Simplified AES cipher for the complete set of 21 equations describing the substitution box. The results are shown in Table 1.

The most important criterion of the presented transformation of the system of multivariate quadratic equations describing the cipher to the QUBO problem was to obtain the QUBO problem with the smallest possible number of binary variables. The obtained results show that reducing the number of equations in the system describing the substitution box reduces the number of required binary variables in the QUBO problem by 55%.

We did some practical experiments, where we tried to use our method to make an algebraic attack on a Simplified AES cipher. We used a D-Wave hybrid solver to solve one round Simplified AES without key expansion. The round keys are permutations of a master key. The equivalent QUBO problem consists of 199 variables. It took 18 minutes of hybrid quantum computation on D-Wave Advantage to obtain the proper result. Because the QUBO problem, equivalent to an algebraic attack on one round Simplified AES with key expansion, consists of more than 300 variables, according to our limitations in available time for computations on D-Wave, we were not able to solve this problem. Even though we think that using more time for computations, larger problems might be solved.

B. TRANSFORMATION OF THE SYSTEM OF MULTIVARIATE POLYNOMIAL EQUATIONS TO THE QUBO PROBLEM FOR AES-128 CIPHER

The transformation to the QUBO problem, discussed in the Simplified AES cipher, was performed for the AES-128 cipher.

The division of the round with the use of intermediate variables was performed analogously to the partition presented in Figures 4 and 5.

The nonlinear layer of the AES-128 encryption algorithm consists of 16 substitution boxes. The number of generated quadratic equations describing the substitution box is 39. This system consists of 120 different monomials of degree two and 1337 monomials.

The system of polynomial equations satisfying the substitution box was searched to find a smaller system. One system of twelve polynomials was found with 64 different monomials of degree two and 667 systems consisting of 13 polynomial equations and 54 different monomials of degree two. Since the primary condition is the number of different monomials of degree two, the system with the lowest number of all monomials, 356, was selected from 667 systems.

The next step was to minimize the system, substituting the equation resulting from the operation xor two, three, four, etc., equations. As a result of this search, a system satisfying the substitution box was found, consisting of 13 polynomial equations, 54 different monomials of degree two, and 268 all monomials.

The obtained minimum system for the substitution box consists of the following equations:

$$eq_0: x_0y_2 + x_0y_4 + x_2y_0 + x_2y_1 + x_3y_1 + x_3y_2 + x_3y_7 + x_4y_3 + x_4y_4 + x_5y_1 + x_5y_2 + x_5y_3 + x_5y_7 + x_6y_1 + x_6y_2 + x_6y_3 + x_0 + x_2 + x_4 + x_7 + y_0 + y_5 = 0,$$

$$eq_1: x_0y_3 + x_0y_5 + x_2y_0 + x_2y_1 + x_2y_5 + x_3y_0 + x_3y_3 + x_3y_7 + x_4y_0 + x_4y_3 + x_4y_4 + x_5y_0 + x_6y_3 + x_6y_4 + x_6y_5 + x_7y_1 + x_7y_5 + x_7y_7 + x_2 + x_4 + y_0 = 0,$$

$$eq_2: x_0y_0 + x_0y_2 + x_0y_4 + x_0y_6 + x_1y_5 + x_1y_7 + x_2y_0 + x_3y_1 + x_3y_2 + x_4y_0 + x_4y_1 + x_4y_7 + x_5y_0 + x_5y_3 + x_6y_4 + x_7y_4 + x_5 + y_1 + y_7 = 0,$$

$$eq_3: x_0y_5 + x_1y_1 + x_3y_0 + x_4y_0 + x_4y_1 + x_4y_3 + x_4y_6 + x_5y_6 + x_6y_1 + x_6y_2 + x_6y_3 + x_7y_2 + x_7y_4 + x_1 + x_4 + x_5 + x_6 + x_7 + y_4 = 0,$$

TABLE 2. Results of Transformation of the System of Multivariate Quadratic Equations Describing the AES-128 Cipher to the QUBO Problem for the Three Sets of Equations Describing the Substitution Box.

The number of equations in the system describing the substitution box.	39	13	13
The number of different monomials of degree two in the system of equations describing the cipher.	120	54	54
The number of all monomials in the system of equations describing the cipher.	1337	356	268
The number of equations describing the cipher.	10488	5288	5288
The number of binary variables in the cipher description system.	4544	4544	4544
The number of variables used during linearization.	24000	10800	10800
The number of binary variables after linearization.	32384	19184	19184
The number of binary variables for the value of k_i.	40082	15282	14682
The number of binary variables in the obtained QUBO problem.	68626	30626	30026

$$eq_4 : x_0y_0 + x_0y_2 + x_0y_4 + x_0y_5 + x_3y_3 + x_3y_4 + x_4y_4 + x_5y_0 + x_5y_1 + x_6y_1 + x_6y_5 + x_7y_2 + x_7y_5 + x_0 + x_1 + x_2 + x_5 + x_6 + x_7 + y_5 + y_6 + y_7 = 0,$$

$$eq_5 : x_0y_0 + x_0y_3 + x_0y_5 + x_2y_0 + x_2y_2 + x_2y_4 + x_2y_5 + x_3y_2 + x_3y_3 + x_3y_4 + x_3y_5 + x_4y_3 + x_5y_0 + x_5y_4 + x_5y_6 + x_6y_0 + x_6y_1 + x_6y_2 + x_0 + y_1 + y_2 = 0,$$

$$eq_6 : x_0y_3 + x_0y_5 + x_0y_7 + x_2y_4 + x_2y_6 + x_2y_7 + x_3y_2 + x_3y_6 + x_4y_4 + x_5y_3 + x_5y_7 + x_6y_4 + x_6y_5 + x_7y_1 + x_7y_7 + x_0 + x_4 + x_5 + y_0 + y_3 = 0,$$

$$eq_7 : x_0y_4 + x_0y_7 + x_1y_5 + x_2y_1 + x_2y_6 + x_2y_7 + x_3y_1 + x_3y_4 + x_4y_0 + x_4y_4 + x_4y_5 + x_4y_7 + x_5y_4 + x_5y_5 + x_6y_4 + x_7y_2 + x_7y_7 + x_3 + x_5 + x_6 + y_0 + y_5 = 0,$$

$$eq_8 : x_0y_0 + x_0y_2 + x_0y_7 + x_2y_0 + x_2y_3 + x_2y_4 + x_3y_4 + x_3y_6 + x_4y_4 + x_5y_1 + x_6y_0 + x_6y_1 + x_6y_2 + x_6y_4 + x_7y_2 + x_7y_3 + x_6 + y_1 + y_2 + y_5 + y_7 + 1 = 0,$$

$$eq_9 : x_0y_0 + x_0y_4 + x_0y_6 + x_0y_7 + x_2y_7 + x_3y_3 + x_4y_0 + x_4y_1 + x_4y_3 + x_4y_6 + x_5y_4 + x_5y_6 + x_5y_7 + x_6y_1 + x_6y_3 + x_6y_4 + x_7y_1 + x_7y_3 + x_2 + y_3 + y_5 = 0,$$

$$eq_{10} : x_0y_2 + x_0y_7 + x_1y_5 + x_1y_7 + x_2y_0 + x_2y_2 + x_3y_4 + x_3y_7 + x_4y_2 + x_4y_3 + x_5y_4 + x_7y_2 + x_7y_5 + x_7y_7 + x_1 + x_7 + y_3 + y_5 + y_7 + 1 = 0,$$

$$eq_{11} : x_2y_3 + x_2y_4 + x_3y_4 + x_4y_0 + x_4y_1 + x_5y_1 + x_5y_5 + x_6y_2 + x_6y_5 + x_7y_1 + x_0 + x_4 + x_5 + x_6 + y_4 + y_5 + y_6 + 1 = 0,$$

$$eq_{12} : x_0y_4 + x_0y_6 + x_1y_7 + x_2y_1 + x_2y_4 + x_2y_5 + x_2y_6 + x_3y_0 + x_3y_2 + x_3y_4 + x_3y_6 + x_4y_1 + x_4y_3 + x_5y_0 + x_5y_1 + x_5y_2 + x_6y_5 + x_4 + y_1 + y_5 + y_7 = 0,$$

where x_i is the S-box input bits and y_i is the S-box output bits.

A complete transformation to the QUBO problem was performed for the three polynomial equation systems that satisfied the substitution box. The results are presented in Table 2.

The number of binary variables of the QUBO problem for the AES-128 cipher with the over-defined system of polynomial equations describing the substitution box is 68626. Due to the search for a smaller system for the substitution box, the number of necessary binary variables of the QUBO problem was reduced to 30626, which gives an improvement of 55%. Another minimization of the system of polynomials satisfying the substitution box to a lesser extent reduced the number of binary variables required in the QUBO problem to 30026, i.e., by 2%.

V. CONCLUSION

In this paper, we presented an algebraic attack on block ciphers using quantum annealing. We showed how to transform a system of equations describing block cipher to the QUBO problem. What is essential, in a similar way, one can transform stream cipher or other symmetric primitive to the QUBO problem. Such a problem may then be solved using quantum annealing, for example, using a D-Wave computer. Because the complexity of solving the QUBO problem depends on the number of variables, we aimed to minimize the number of variables in the obtained QUBO problem. We made such minimization in the AES-128 cipher, searching for a minimal system of equations with the minimum number of different monomials of degree two, satisfying the substitution box.

What is more, such a system was then modified by XORing some of the equations, where finally, we obtained a smaller number of all monomials in the system describing the substitution box. These methods let us decrease the number of variables of complete AES-128 cipher from 68,626 to 30,026 binary variables, which gave us 56, 25% of variables less than using complete system of equations describing substitution box. It is worth noting that it means that AES-128 is much easier to break using quantum annealing than the factorization problem and the discrete logarithm of a similar level of security. For example, using quantum annealing to factorize a 3072 bit long RSA integer requires a QUBO problem of about 2,360,000 variables. Unfortunately, nowadays asymptotic time of execution of the QUBO problem using quantum annealing is not known. It seems that problems of such sizes as the QUBO problem for AES-128 will not be solved using quantum annealing in the following years. However, some efficient algorithm for solving QUBO problems using the general-purpose quantum computer may be found in the near future.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.

[2] T. Greene. Google reclaims quantum computer crown with 72 qubit processor. 2018. [Online]. Available: https://thenextweb.com/artificial-intelligence/2018/03/06/google-reclaims-quantum-computer-crown-with-72-qubit-processor/

[3] R. Dridi and H. Alghassi, "Prime factorization using quantum annealing and computational algebraic geometry," *Sci. Rep.*, vol. 7, 2017, Art. no. 43048.

[4] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, "Quantum annealing for prime factorization," *Sci. Rep.*, vol. 8, no. 1, pp. 1–9, 2018.

[5] B. Wang, F. Hu, H. Yao, and C. Wang, "prime factorization algorithm based on parameter optimization of ising model," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, 2020.

[6] M. Wroński, "Practical solving of discrete logarithm problem over prime fields using quantum annealing," *Cryptology ePrint Archive, Rep. 2021/527*, 2021, [Online]. Available: https://eprint.iacr.org/2021/527

[7] G. Bard, *Algebraic cryptanalysis*. Berlin, Germany: Springer Science & Business Media, 2009.

[8] N. T. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2002, pp. 267–287.
[9] N. T. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2003, pp. 345–359.
[10] N. T. Courtois and G. V. Bard, "Algebraic cryptanalysis of the data encryption standard," in *Proc. IMA Int. Conf. Cryptogr. Coding*, 2007, pp. 152–169.
[11] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, "Efficient algorithms for solving overdefined systems of multivariate polynomial equations," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2000, pp. 392–407.
[12] A. Shamir and A. Kipnis, "Cryptanalysis of the HFE public key cryptosystem," in *Proc. 19th Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 1999, pp. 19–30.
[13] P. L. Hammer and S. Rudeanu, *Boolean methods in operations research and related areas*. Berlin, Germany: Springer Science & Business Media, 2012.
[14] I. G. Rosenberg, "Reduction of bivalent maximization to the quadratic case," *Cahiers Centre Etudes Rech. Oper.*, vol. 17, pp. 71–74, 1975.
[15] D. O'Malley and V. V. Vesselinov, "ToQ. jl: A high-level programming language for d-wave machines based on julia," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2016, pp. 1–7.
[16] A. Borle and S. J. Lomonaco, "Analyzing the quantum annealing approach for solving linear least squares problems," in *WALCOM: Algorithms and Computation*, G. K. Das, P. S. Mandal, K. Mukhopadhyaya, and S.-I. Nakano, Eds. Berlin, Germany: Springer, 2019, pp. 289–301.
[17] T. H. Chang, T. C. Lux, and S. S. Tipirneni, "Least-squares solutions to polynomial systems of equations with quantum annealing," *Quantum Inf. Process.*, vol. 18, no. 12, pp. 1–17, 2019.
[18] C. C. Chang, A. Gambhir, T. S. Humble, and S. Sota, "Quantum annealing for systems of polynomial equations," *Sci. Rep.*, vol. 9, no. 1, 2019, Art. no. 10258.
[19] M. A. Musa, E. F. Schaefer, and S. Wedig, "A simplified aes algorithm and its linear and differential cryptanalyses," *Cryptologia*, vol. 27, no. 2, pp. 148–177, 2003.
[20] N. Sklavos, "Book review: Stallings, w. cryptography and network security: Principles and practice," *Inf. Secur. J.: A Global Perspective*, vol. 23, no. 1/2, pp. 49–50, 2014.

ELŻBIETA BUREK received the MSc degree in cartography, in 2012, and the BSc degree in computer science, specialization in cryptology, in 2019, both from the Military University of Technology, Warsaw, Poland. She is currently an assistant with the Military University of Technology in Warsaw. Her research interests include algebraic attacks on blocks ciphers.

MICHAŁ WROŃSKI received the PhD degree in computer science, specialization cryptology from the Military University of Technology, Warsaw, Poland, in 2018. He is currently an assistant professor with the Military University of Technology in Warsaw. His research interests include elliptic curve cryptography, especially isogeny-based cryptography, and the application of quantum annealing to cryptology.

KRZYSZTOF MAŃK received the MSc degree in computer science, specialization cryptology from the Military University of Technology, Warsaw, Poland, in 1999. He is currently an assistant with the Military University of Technology in Warsaw. His research interests include random number generators and stream ciphers.

MICHAŁ MISZTAL received the PhD degree in computer science, specialization cryptology from the Military University of Technology, Warsaw, Poland, in 2008. He is currently an assistant professor with the Military University of Technology in Warsaw. His research interests include symmetric cryptography, especially construction and algebraic attacks on block ciphers.