Tatra Mt. Math. Publ. **91** (2025), 23–50
DOI: 10.2478/tmmp-2025-0006

TATRA MOUNTAINS
Mathematical Publications

CONVERTING OF SIMON CIPHER MULTIVARIATE POLYNOMIAL EQUATIONS TO QUBO PROBLEM

Elżbieta Burek

Cybernetics Faculty, Military University of Technology, Warsaw, POLAND

ABSTRACT. The use of quantum annealing in the cryptanalysis of symmetric cryptography is a new idea based on the concept of algebraic attacks. This paper shows how to describe the Simon cipher as a system of multivariate polynomial equations so that the obtained optimization problem in the form of QUBO consists of as small number of binary variables as possible.

According to our calculations, the use of quantum annealing to an algebraic attack on the Simon128/128 cipher, since the QUBO problem consists of 27,270 binary variables, is more effective than the same attack on the AES128 cipher, for which the QUBO problem includes 29,770 binary variables.

1. Introduction

The dynamically growing value of the Internet of Things market makes it one of the key branches of technological development. The Internet of Things is a system that enables devices to connect to a computer network and to process data on their own with the help of special sensors. However, transmitted data must be secured. Since the resources of these devices are limited, special cryptographic algorithms, called lightweight ciphers, have been designed. One such cipher is the Simon cipher, designed in 2013 by the NSA. In [6], and [7], the best complexities of linear and differential attacks for all versions of the SIMON cipher have been presented. In [10], an attempt at an algebraic attack is also presented.

The Shor [12] algorithm was initiated using quantum computing to factorize numbers. The Grover algorithm is a well-known quantum algorithm that can be

© 2025 Mathematical Institute, Slovak Academy of Sciences.
2020 Mathematics Subject Classification: 68U99.
Keywords: cryptanalysis, Simon cipher, quantum annealing, D-wave advantage.
Supported by the Grant DOB/002/RON/ID1/2018 by The National Centre for Research and Development.
Licensed under the Creative Commons BY-NC-ND 4.0 International Public License.

used in the cryptanalysis of symmetric ciphers. An attack using the Grover algorithm is an exhaustive search where quantum implementation of a given cipher is used. The existence of this attack reduces the security of symmetric ciphers to $O(2^{m/2})$, where m is the key length. Moreover, this attack targets general-purpose quantum computers that currently have too little resources to perform this attack. For example, the most powerful general-purpose quantum computer was built by IBM and had 127 qubits. Therefore, quantum computers using quantum annealing get more and more interest. They have much more resources, for example the D-Wave Advantage computer has $5,760$ qubits. This quantum annealer allows for solving general problems (presented in Ising or QUBO form) with up to $1,000,000$ variables and dense problems with up to $20,000$ variables.

The QUBO (Quadratic Unconstrained Boolean Optimization) problem is a combinatorial optimization problem in which the cost function $f(x)$ is defined on an n-dimensional binary vector space $\mathbb{B}^n$ to $\mathbb{R}$, which is as follows:

$$\text{QUBO} : \min \ f(x) = x^t Q x, \tag{1}$$

where x is a vector of binary variables and Q is an upper triangular matrix of real weights. Since for binary variables hold $x^2 = x$, the cost function can be represented in the form:

$$\text{QUBO} : \min \ f(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j. \tag{2}$$

One of the first cryptographic problems, presented in the form of the QUBO problem, was the problem of integer factorization, which is the problem of asymmetric cryptography. This transformation was proposed in [8]. The next work [13] examined a discrete logarithm problem, which is also the problem of asymmetric cryptography.

In [3], it was presented how to solve a linear system of multivariate polynomial equations using quantum annealing. On this basis and using the idea of algebraic attacks, in [5], we presented a method of transforming the problem of solving the system of multivariate polynomial equations, describing the AES cipher, into an optimization problem in the form of QUBO. The general idea of this method is as follows.

Firstly, one has to obtain a system of multivariate polynomial equations f_i, defined over the field $GF(q)$, representing a given cipher. Then each equation f_i is transformed into an equation with Boolean variables and integer coefficients of the form $f_i' = f_i - q \cdot k_i$, where k_i is an integer and $k_i \leq \lfloor \frac{f_{i_{\max}}}{q} \rfloor$. The value of $f_{i_{\max}}$ is the maximum value that polynomial f_i can have when all of its binary variables have the value of one. Moreover, the value of k_i has to be written as

the sum of Boolean variables k_{i_j}, according to the following equation:

$$k_i = \sum_{j=0}^{bl(k_{i_{\max}})-2} 2^j k_{i_j} + \left(k_{i_{\max}} - 2^{bl(k_{i_{\max}})-1} + 1\right) \cdot k_{i_{bl(k_{i_{\max}})-1}}, \tag{3}$$

where $bl(x)$ is the bit-length of integer x and $k_{i_{\max}} = \lfloor \frac{f_{i_{\max}}}{q} \rfloor$.

Since the transformation of the linear equations system into the QUBO problem is executed, in the next step, the linearization of the equations f'_i is performed, obtaining $f'_{i_{\text{lin}}} = \text{lin}(f'_i)$, where lin denotes the linearization operation. The problem of linearization of multivariate polynomials is NP-hard. However, this method does not require the optimal solution. The Rosenberg [11] method was used to perform the linearization, replacing each quadratic monomial with a new auxiliary binary variable. Since the target QUBO problem is in an unconstrained form, an appropriate penalty must be introduced in order to relate the new variable with the replaced monomial. For example, when replacing the quadratic monomial $x_i x_j$ with x_k, the following substitution is performed:

$$x_i x_j \to x_k + 2\big(x_i x_j - 2x_k(x_i + x_j) + 3x_k\big), \tag{4}$$

where $2(x_i x_j - 2x_k(x_i + x_j) + 3x_k)$ is a penalty. If $x_i x_j = x_k$, then the penalty is zero. Otherwise, if $x_i x_j \neq x_k$, then $x_i x_j < x_k + 2\big(x_i x_j - 2x_k(x_i + x_j) + 3x_k\big)$, where the penalty is so high that such a solution should be rejected during the search for the minimum of a cost function. If the replaced monomial has a degree d, then $d-1$ substitutions must be executed. The cost of each reduction is one new binary 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 reduction. In this method, successive substitutions during linearization were performed in the order of the appearance of successive quadratic monomials. After all substitutions the penalty denoted as Pen_{lin}, is multiplied by the large positive constant M and added to the cost function of the QUBO problem.

The last step is to find the sum of squares of all the polynomials $f'_{i_{\text{lin}}}$, yielding $\sum_i (f'_{i_{\text{lin}}})^2$. Finally, the problem in the QUBO form was obtained as

$$\sum_i (f'_{i_{\text{lin}}})^2 + M \cdot \text{Pen}_{\text{lin}} - C, \tag{5}$$

where C is a constant appearing in the polynomial $\sum_i (f'_{i_{\text{lin}}})^2 + M \cdot \text{Pen}_{\text{lin}}$. The binary variables coefficients in equation (5) correspond with the elements of Q matrix in equation (2). The Q matrix is given to D-Wave System.

The number of binary variables in the final QUBO problem depends on the size of the multivariate polynomial equation system describing the cipher and the degree of these equations. However, the size of the system and the degree

of equations depend on the structure of the cipher, on the round operations and on the algebraic structure over which round operations are defined.

In this paper we try to analyze the process of solving optimization problems by the D-Wave quantum annealer and its impact on the requirements for the equation system representing a given cipher. Moreover, the analysis of the structure of the Simon cipher, the way of describing it using the system of multivariate polynomial equations, and the size of the obtained QUBO problem for all variants of the Simon cipher family will be presented. For example, for the Simon128/128 cipher, an optimization problem in the form of QUBO, consisting of $27,270$ binary variables, was obtained, which turned out to be smaller than the QUBO problem for the AES128 cipher composed of $29,770$ binary variables.

2. D-Wave quantum computer

The D-Wave computer model was designed according to the adiabatic model of quantum computing. It is not a full processor as we understand it today but rather an intelligent memory accelerator designed to solve NP-hard optimization problems using quantum annealing. Quantum annealing, in turn, can be seen as a kind of heuristic search that moves around the solution graph looking for low-lying areas, i.e., performs a random walk described by a parameterized Markov chain. The decision on the direction of the step to be performed in subsequent iterations is based on the transverse field factor, also known as the tunneling rate.

The D-Wave platform consists of two main components:

- a hardware system with QPUs (Quantum Processing Units) that solves problems defined in the Ising model by the physical implementation of the quantum annealing algorithm;
- a conventional Intel server with a user interface, connected to the hardware by control lines and I/O subsystems.

In order to use the D-Wave quantum computer to solve a problem, the problem should be presented as an optimization problem. The mathematical representation of the optimization problem is the cost function, which expresses the energy of the system and whose lowest value represents the best solution. For the QPU solver, energy is a function of the binary variables representing its qubits. In some cases, each low energy state is an acceptable solution to the original problem. However, there are problems for which only the optimal solution in the solution space is acceptable. The optimal solution has minimum global energy and it is the solution from the ground state. Finding such a solution is necessary when using quantum annealing for algebraic attacks to recover the correct key.

2.1. QPU topology of D-Wave quantum computer

The QPU of the D-Wave system is a network of qubits connected using couplers. However, it is not a complete graph. As of today, there are two topologies available, according to which qubits are connected:

- Chimera — used in D-Wave 2000Q systems and earlier generations of QPU units;
- Pegasus — used in D-Wave Advantage systems.

Fig. 1 and 2 show the QPU as a graph for the Chimera and Pegasus topologies, respectively. The qubits in these graphs are presented as horizontal and vertical loops with couplers between them.

In the case of the Chimera topology, two types of couplers are distinguished:

- internal couplers that connect pairs of qubits of opposite orientation (perpendicular loops), which in Fig. 1 are marked with green circles;
- external couplers, connecting pairs of collinear qubits, i.e., pairs of parallel qubits on the same row or column; in Fig. 1 the external couplers are shown as blue circles that connect horizontal qubits to adjacent horizontal qubits.

In the Chimera topology, the nominal length of qubit connections is 4, which means that each qubit is connected to 4 orthogonal qubits employing internal couplers. Additionally, the joins of qubits have a degree of 6, which means that each qubit is linked to 6 different qubits.

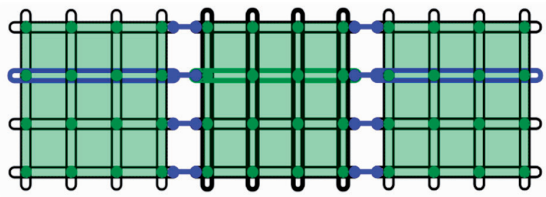

FIGURE 1. The couplers of qubits in Chimera topology, source: [1].

In the case of the Pegasus topology, there are three types of couplers:

- internal couplers, connecting pairs of orthogonal qubits, where each qubit is connected to 12 other qubits, in Fig. 2 a green, the vertical qubit is coupled to 12 horizontal qubits, marked with a bold black line;
- external couplers that connect similarly aligned, adjacent vertical qubits with adjacent vertical qubits, and the horizontal ones with adjacent horizontal qubits, in Fig. 2 the vertical qubit, marked in green, has been coupled with two adjacent vertical qubits marked in blue;

- odd couplers connecting similarly aligned pairs of qubits, for example, in Fig. 2, a green vertical qubit has been coupled to a red vertical qubit by an odd coupler.

Pegasus contains qubit joins of a degree 15 — each qubit is connected to 15 different qubits, and the qubit joins have a nominal length of 12 — each qubit is connected to 12 orthogonal qubits, using internal couplers.

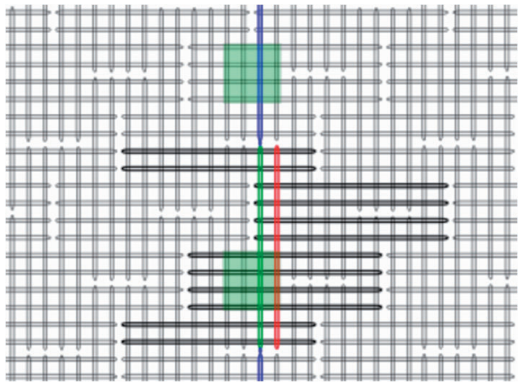

FIGURE 2. The couplers of qubits in Pegasus topology, source: [1].

2.2. Solving optimization problems with the D-Wave quantum computer

The steps of the solving process using the D-Wave computer are as follows:

1. Problem formulation. A given problem should be presented as a combinatorial optimization problem, defining the cost function and its constraints.

2. Converting the problem to a form acceptable to the D-Wave computer. The formulated problem should be mapped to the form of the Ising model or the QUBO problem, which belongs to the category of binary quadratic models (BQM). The native problem of the D-Wave quantum computer is the problem defined in the Ising model, according to the following equation:

$$H(s) = \sum_i h_i s_i + \sum_{i<j} J_{i,j} s_i s_j. \tag{6}$$

This problem is directly implemented in the D-Wave hardware system. However, if the defined problem is not native, the D-Wave system transforms the problem to the Ising model first and then performs the next step.

3. Embedding in a QPU. Solving the problem in the D-Wave system requires mapping, which is called embedding, the Ising model graph to the topology graph of a given QPU. Let $G = (V_g, E_g)$ denote the connection structure of the Ising model where the vertices of the set V_g correspond to the variables s_i of the problem, and the edges of the set $E_g = (v_i, v_j)$ correspond to the weights $J_{i,j}$ if they exist ($J_{i,j} \neq 0$). Furthermore, let $H = (V_h, E_h)$ represents the hardware graph of the topology. The problem is to find the smallest possible embedment of graph G in graph H, as shown in Fig. 3, for the Chimera topology. The QPU topology is not a complete graph, so bigger problems often require creating chains. One logical variable is represented by a number of physical qubits linked by a chain. For a qubit chain to represent a logical variable, all of its qubits must return the same value for a given reading. It is achieved by setting the strength of their connecting couplers. For qubits in a chain to return identical values, the strength of the coupling of their edges must be greater than that of the coupling to other qubits. If the chain force is too weak, it breaks, and, as a consequence, incorrect solutions are obtained. On the other hand, setting too strong chain distorts the problem. Additionally, all chains in a given problem should be short and of similar length. During embedding, the weights are also scaled to the ranges acceptable to the physical QPU chip, which is -2 to $+2$ for vertex weights and -1 to $+1$ for edge weights. If the problem in the Ising model has large weight values, they will be scaled to a much smaller range. Therefore, when transforming a given problem into the BQM model, one should also pay attention to the difference between the smallest and the largest coefficients in the Q matrix of the BQM model. For example, let the smallest coefficient be $h_1 = 0,032$ and the largest $h_2 = 3405$. When scaling the coefficients down to the range supported by QPU, the small coefficient is zero because it equals $9,4 \cdot 10^{-6}$.

4. Configuration of the QPU solver. The most important parameter to be determined is the annealing time. According to the adiabatic theory, too short annealing time may cause the system to jump from the ground state to a higher excited state, which makes it impossible to obtain the solution sought, which is the global minimum. However, finding the exact value of the required annealing time is equivalent to finding a solution and is, therefore NP-hard problem.

5. Carrying out calculations. When a D-Wave quantum computer solves a problem, it uses quantum phenomena such as superposition and tunneling to simultaneously test possible solutions and find the best set, i.e., with the lowest system energy.

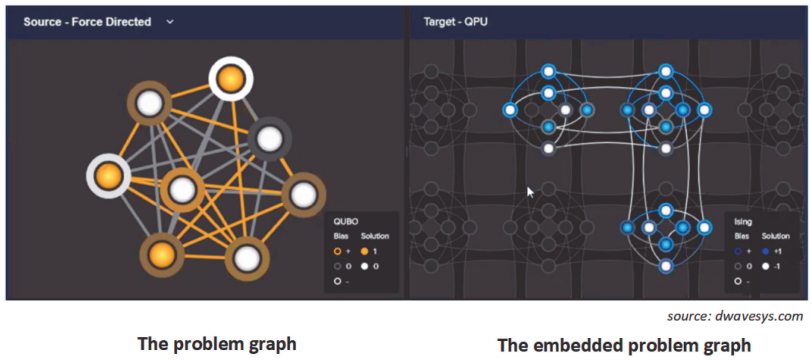

FIGURE 3. Embedding a QUBO problem graph into a hardware graph, source: [1].

3. Requirements for a system of multivariate polynomial equations describing a given cipher

Taking into account the transformation model of the system of multivariate polynomial equations describing a given cipher proposed in [5] into the QUBO problem as well as the above description of the process of finding solutions using the D-Wave quantum annealer, the requirements were proposed, which should be taken into account while generating the equations system representing a given cipher. Due to the physical structure of the D-Wave quantum computer QPU chip:

- the Q matrix of the obtained QUBO problem should be sparse so that the number and the length of chains created during graph embedding would be small;
- the difference between the maximum and the minimum coefficient in the Q matrix should be as small as possible so that the scaled weights will not be insignificant.

Due to the model of transformation of the system of equations into the QUBO problem:

- the number of different monomials of degree 2 should be as small as possible in order to use the smallest possible number of additional binary variables during the linearization;
- if possible, longer polynomial equations should be used rather than additional equations in the system, which should reduce the number of additional binary variables to represent the value of multiple k_i of each equation.

4. Generating a system of multivariate polynomial equations for the Simon cipher

This chapter presents how the Simon cipher is represented with the multivariate polynomial equation system so that the target QUBO problem counts the requirements mentioned above.

4.1. The Simon cipher

The Simon cipher was designed as a lightweight block cipher with a Feistel structure and, along with the Speck cipher, it was presented in [2]. The Simon cipher is a family of ciphers with various parameters. Let us denote the instance of the Simon cipher with Simon$2n/mn$, where $2n$ is the length of the input block, n is the length of the word, and mn is the length of the key. Fig. 4 shows a diagram of the encryption algorithm of the Simon cipher, and Fig. 5 shows a diagram of the round key generation algorithm, depending on the value of the m parameter. The Simon$2n/mn$ cipher uses the following bitwise operations, performed on n-bit words:

- bitwise XOR operation denoted as $\oplus$,
- bitwise AND operation, denoted as $\&$,
- right and left circular shift by j bits, denoted as $\gg j$ and $\ll j$, respectively.

The bitwise AND operation is the nonlinear operation, while the remaining operations are linear. The number of rounds is denoted with the letter T and ranges from 32 to 72, depending on the instance of the Simon cipher.

For a given round key

$$kr \in GF(2)^n,$$

the round function of the encryption algorithm of the Simon$2n/mn$ cipher is a mapping of

$$R_{kr} : GF(2)^n \times GF(2)^n \to GF(2)^n \times GF(2)^n,$$

defined in the following way:

$$R_{kr}(x_{i+1}, y_{i+1}) = \big(y_i \oplus F(x_i) \oplus kr_i, x_i\big), \tag{7}$$

where kr_i is the round key, i is the number of rounds, and function F is defined as follows:

$$F(x_i) = \big((x_i \ll 1) \,\&\, (x_i \ll 8)\big) \oplus (x_i \ll 2).$$

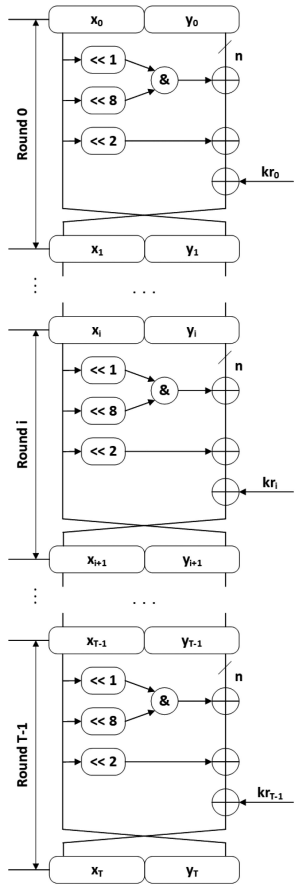

FIGURE 4. Structure of the encryption algorithm of the Simon$2n/mn$ cipher, source: based on [2].

Each instance of the Simon$2n/mn$ cipher uses the known principle of the data processing direction, according to the classic Feistel network, where only a half of the input state is processed in a given round. As shown in Fig. 4, according to Feistel's idea, the $2n$-bit input state to the i round is split into two n-bit words, where the most significant n bits are the word x_i, and the n the least significant bits are the word y_i. The x_i word is processed with the F function, which consists of a parallel left circular shift by 1, 8 and 2 bits. The bitwise AND operation is performed on the results of the first two shifts, the result of which is XORed with the result of the third shift. In this way, the value of the function $F(x_i)$ is obtained, which is then XORed with the ith round key and with the y_i word, forming the x_{i+1} word of the next round. On the other hand, the y_{i+1} word is the unprocessed x_i word.

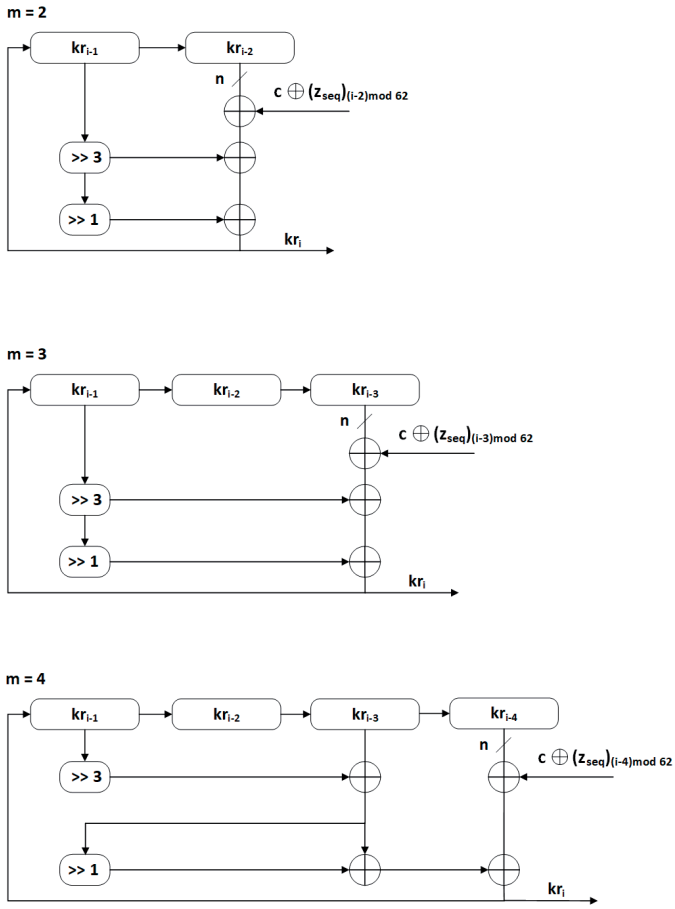

FIGURE 5. Structure of the round key generation algorithm of the Simon $2n/mn$ cipher, source: based on [2].

The key K of the Simon$2n/mn$ cipher is divided into m, n-bit round keys as follows: $K = [kr_{m-1}, \ldots, kr_0]$, where $kr_i \in GF(2)^n$. Based on the key K, the remaining $T - m$ round keys are generated. Fig. 5 shows the algorithm for generating round keys of the Simon cipher for three possible values of the m parameter. The method of determining the ith round key is identical for $m=2$ and $m = 3$: the previous key, kr_{i-1}, is circular shifted to the right by 3 and

4 bits, and then the obtained results are XORed with the round key kr_{i-m} and the constant $c \oplus (z_{seq})_{(i-m) \bmod 62}$, yielding the kr_i round key. In the case when $m = 4$, the kr_{i-1} key is circular shifted to the right only once, by 3 bits, and then XORed with the kr_{i-3} key. The obtained result is XORed with its circular shift to the right by 1 bit, and then it is XORed with the kr_{i-4} key and the constant $c \oplus (z_{\text{seq}})_{(i-m) \bmod 62}$, where:

- c is an n-bit constant number, equal to $2^n - 4 = 0xfff \cdots fc$;
- $(z_{\text{seq}})_{(i-m) \bmod 62}$ is a $(i - m) \bmod 62$th bit of the sequence z_{seq}.

4.2. Generating equations describing the encryption algorithm of the Simon cipher

Due to the bit character of all operations of the Simon cipher, the natural algebraic structure over which the polynomial equations of the Simon cipher have been presented is the binary field $GF(2)$. Let's assume that binary variables represent the round key bits, and polynomial equations are generated for each intermediate state bit, each round of the Simon cipher.

Let n-bit words a and b be presented as a string of bits of the form:

$$a = a_{n-1}, a_{n-2}, \ldots, a_1, a_0,$$
$$b = b_{n-1}, b_{n-2}, \ldots, b_1, b_0,$$

where a_j and b_j are the j-th bit of word. Left circular shift of word a by 1, 2, and 8 bits is performed as a cyclic bit shift the string according to equations (8), (9) and (10), respectively.

$$a \lll 1 = a_{n-2}, a_{n-3}, \ldots, a_1, a_0, a_{n-1}, \tag{8}$$

$$a \lll 2 = a_{n-3}, a_{n-4}, \ldots, a_1, a_0, a_{n-1}, a_{n-2}, \tag{9}$$

$$a \lll 8 = a_{n-9}, a_{n-10}, \ldots, a_{n-6}, a_{n-7}, a_{n-8}. \tag{10}$$

The bitwise XOR and AND operations of the two bits a_j and b_j can be realized as addition and multiplication over $GF(2)$. Hence, the XOR/AND operation for two n-bit words a and b, is performed as addition/multiplication of the corresponding bits of the words, as shown by the equation (11) for the XOR operation and by the equation (12) for the AND operation.

$$a \oplus b = a_{n-1} + b_{n-1}, a_{n-2} + b_{n-2}, \ldots, a_1 + b_1, a_0 + b_0, \tag{11}$$

$$a \,\&\, b = a_{n-1}b_{n-1}, a_{n-2}b_{n-2}, \ldots, a_1b_1, a_0b_0. \tag{12}$$

During the generation of polynomial equations describing the encryption algorithm of the Simon cipher its structure was used, connecting adjacent rounds in pairs, and reducing the number of equations by half. Fig. 6 shows how to connect two adjacent rounds with variables representing intermediate states. The words x_i, y_i, x_{i+1} and y_{i+1} are intermediate states between double rounds

and are represented by additional binary variables. The word x_i, in blue, is processed by the function $F()$ whose operations, in Fig. 6, are denoted by a green dashed line. At the same time, the binary variables representing the word x_i form the right word of the round $i+1$ denoted by the blue dashed line. Similarly, the word y_{i+1}, represented by additional binary variables, is simultaneously the left word of round $i+1$, denoted by a red, dotted line, and the result of processing the y_i word of round i. Based on round i, the y_{i+1} word is defined as $y_{i+1} = F(x_i) \oplus y_i \oplus kr_i$, while round $i+1$ defines the word x_{i+1} as $x_{i+1} = F(y_{i+1}) \oplus x_i \oplus kr_{i+1}$.

A system of polynomial equations was constructed based on the defined relationships, describing the double round of the encryption algorithm of the Simon$2n/mn$ cipher. Let us present the words of the intermediate states as a string of bits of the following form:

$$x_i = x_{i_{n-1}}, x_{i_{n-2}}, \ldots, x_{i_1}, x_{i_0},$$

$$y_i = y_{i_{n-1}}, y_{i_{n-2}}, \ldots, y_{i_1}, y_{i_0},$$

$$x_{i+1} = x_{i+1_{n-1}}, x_{i+1_{n-2}}, \ldots, x_{i+1_1}, x_{i+1_0},$$

$$y_{i+1} = y_{i+1_{n-1}}, y_{i+1_{n-2}}, \ldots, y_{i+1_1}, y_{i+1_0}.$$

Hence, this system consists of $2n$ equations of the form (13), the first n of which describe the word x_{i+1}, and the last n of which describe the word y_{i+1}.

$$\begin{aligned}
&f_1 : y_{i+1_{n-2}} y_{i+1_{n-9}} + y_{i+1_{n-3}} + x_{i_{n-1}} + kr_{i+1_{n-1}} - x_{i+1_{n-1}} = 0,\\
&f_2 : y_{i+1_{n-3}} y_{i+1_{n-10}} + y_{i+1_{n-4}} + x_{i_{n-2}} + kr_{i+1_{n-2}} - x_{i+1_{n-2}} = 0,\\
&\vdots\\
&f_{n-2} : y_{i+1_1} y_{i+1_{n-6}} + y_{i+1_0} + x_{i_2} + kr_{i+1_2} - x_{i+1_2} = 0,\\
&f_{n-1} : y_{i+1_0} y_{i+1_{n-7}} + y_{i+1_{n-1}} + x_{i_1} + kr_{i+1_1} - x_{i+1_1} = 0,\\
&f_n : y_{i+1_{n-1}} y_{i+1_{n-8}} + y_{i+1_{n-2}} + x_{i_0} + kr_{i+1_0} - x_{i+1_0} = 0,\\
&f_{n+1} : x_{i_{n-2}} x_{i_{n-9}} + x_{i_{n-3}} + y_{i_{n-1}} + kr_{i_{n-1}} - y_{i+1_{n-1}} = 0,\\
&f_{n+2} : x_{i_{n-3}} x_{i_{n-10}} + x_{i_{n-4}} + y_{i_{n-2}} + kr_{i_{n-2}} - y_{i+1_{n-2}} = 0,\\
&\vdots\\
&f_{2n-2} : x_{i_1} x_{i_{n-6}} + x_{i_0} + y_{i_2} + kr_{i_2} - y_{i+1_2} = 0,\\
&f_{2n-1} : x_{i_0} x_{i_{n-7}} + x_{i_{n-1}} + y_{i_1} + kr_{i_1} - y_{i+1_1} = 0,\\
&f_{2n} : x_{i_{n-1}} x_{i_{n-8}} + x_{i_{n-2}} + y_{i_0} + kr_{i_0} - y_{i+1_0} = 0,
\end{aligned} \tag{13}$$

where i is the number of the first round of the pair. Considering the degree of generated equations, it can be seen that all equations have degree 2, with each

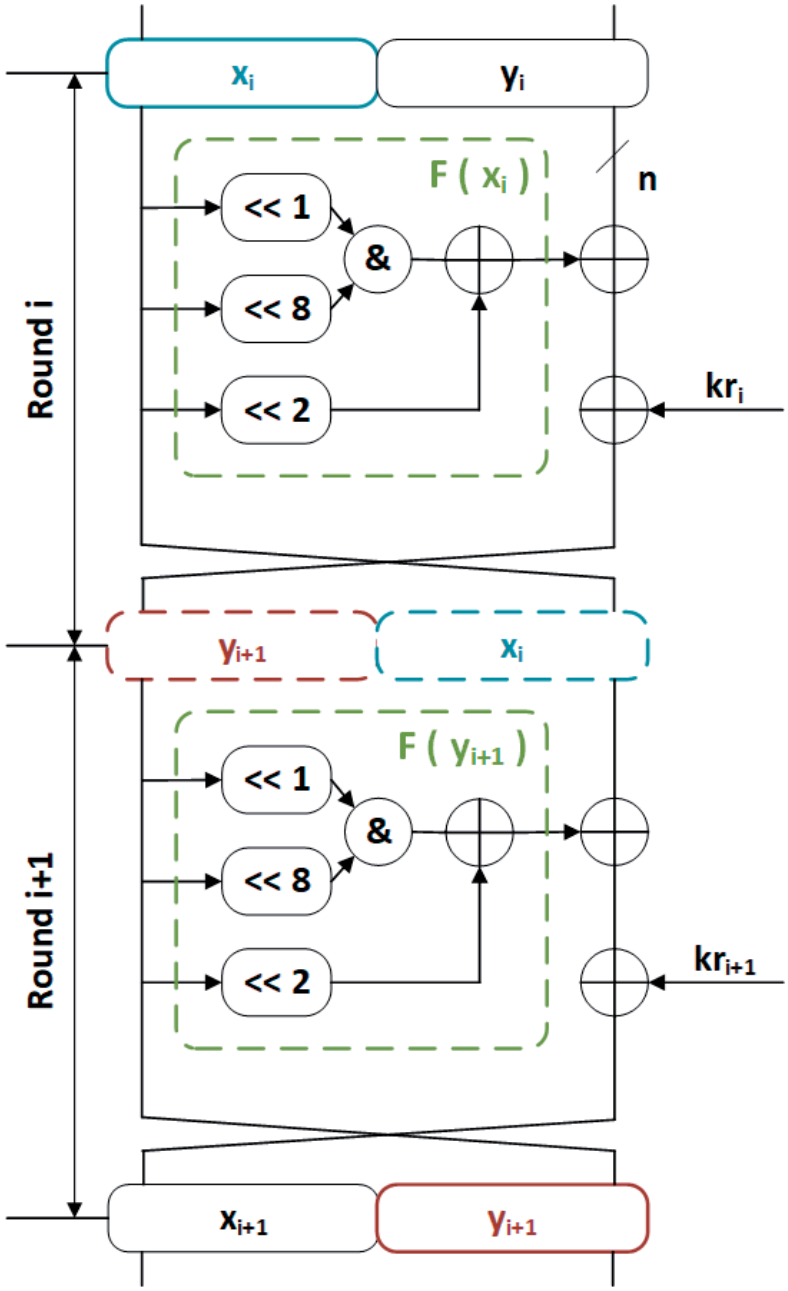

FIGURE 6. The scheme of connecting two adjacent rounds of the Simon$2n/mn$ cipher.

equation having exactly one monomial of degree 2. Therefore, during linearization, for one double round of the encryption algorithm, $2n$ additional variables are required.

Extending the above description to the entire encryption algorithm, in Fig. 7, the round keys and intermediate states between double rounds, represented by binary variables, are denoted by red color. The number of binary variables needed to create a system of polynomial equations for the Simon cipher is the sum of the number of variables for intermediate states (words x_1 to $x_{(T-2)/2}$ and y_1 to $y_{(T-2)/2}$) and of the number of binary variables for round keys (words from kr_m to kr_{T-1}), hence this sum is equal to

$$(T-2)n + (T-m)n.$$

In Fig. 7 the known bits of plaintext and the corresponding ciphertext are denoted as words PT_1, PT_0, CT_1 and CT_0. Considering the knowledge of these bits, the equations system for the first pair of rounds takes the form of equations (14), of which the first n equations are quadratic equations and the last n are linear equations. Hence, linearizing the first pair of rounds equations requires the definition of n additional binary variables.

$$\begin{aligned}
&f_1 : y_{1_{n-2}} y_{1_{n-9}} + y_{1_{n-3}} + PT_{1_{n-1}} + kr_{1_{n-1}} - x_{1_{n-1}} = 0,\\
&f_2 : y_{1_{n-3}} y_{1_{n-10}} + y_{1_{n-4}} + PT_{1_{n-2}} + kr_{1_{n-2}} - x_{1_{n-2}} = 0,\\
&\vdots\\
&f_{n-2} : y_{1_1} y_{1_{n-6}} + y_{1_0} + PT_{1_2} + kr_{1_2} - x_{1_2} = 0,\\
&f_{n-1} : y_{1_0} y_{1_{n-7}} + y_{1_{n-1}} + PT_{1_1} + kr_{1_1} - x_{1_1} = 0,\\
&f_n : y_{1_{n-1}} y_{1_{n-8}} + y_{1_{n-2}} + PT_{1_0} + kr_{1_0} - x_{1_0} = 0,\\
&f_{n+1} : PT_{1_{n-2}} PT_{1_{n-9}} + PT_{1_{n-3}} + PT_{0_{n-1}} + kr_{0_{n-1}} - y_{1_{n-1}} = 0,\\
&f_{n+2} : PT_{1_{n-3}} PT_{1_{n-10}} + PT_{1_{n-4}} + PT_{0_{n-2}} + kr_{0_{n-2}} - y_{1_{n-2}} = 0,\\
&\vdots\\
&f_{2n-2} : PT_{1_1} PT_{1_{n-6}} + PT_{1_0} + PT_{0_2} + kr_{0_2} - y_{1_2} = 0,\\
&f_{2n-1} : PT_{1_0} PT_{1_{n-7}} + PT_{1_{n-1}} + PT_{0_1} + kr_{0_1} - y_{1_1} = 0,\\
&f_{2n} : PT_{1_{n-1}} PT_{1_{n-8}} + PT_{1_{n-2}} + PT_{0_0} + kr_{0_0} - y_{1_0} = 0,
\end{aligned} \tag{14}$$

The last pair of rounds, due to the knowledge of the ciphertext is described by the system of equations of the form (15), of which the first n equations are linear equations, and the last n are quadratic equations, with one quadratic

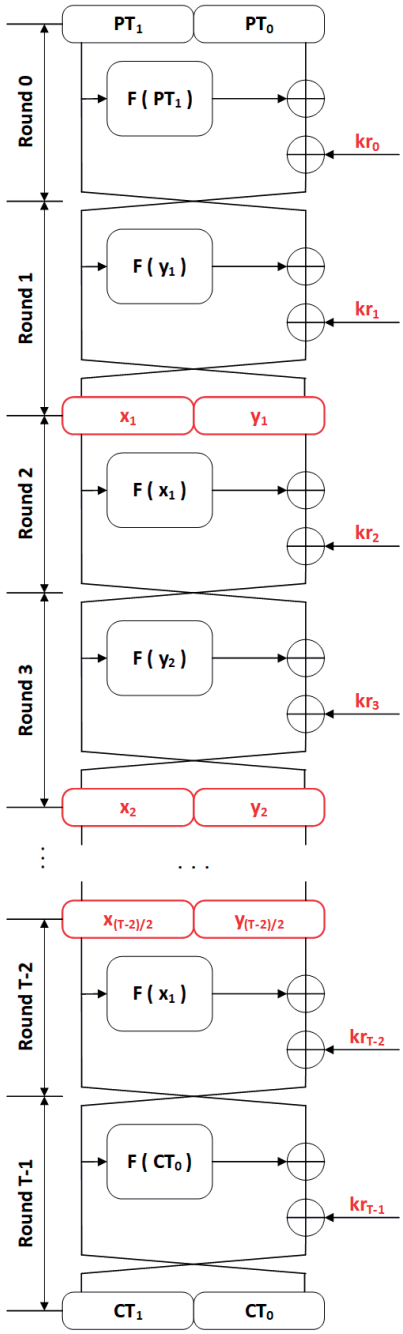

FIGURE 7. Split of Simon$2n/mn$ cipher by using additional variables.

monomial in each equation. When linearizing the equations representing the last pair of rounds, n additional binary variables are required.

$$
\begin{aligned}
&f_1 : CT0_{n-2}CT0_{n-9} + CT0_{n-3} + x_{(T-2)/2_{n-1}} + kr_{T-1_{n-1}} - CT1_{n-1} = 0,\\
&f_2 : CT0_{n-3}CT0_{n-10} + CT0_{n-4} + x_{(T-2)/2_{n-2}} + kr_{T-1_{n-2}} - CT1_{n-2} = 0,\\
&\vdots\\
&f_{n-2} : CT0_1 CT0_{n-6} + CT0_0 + x_{(T-2)/2_2} + kr_{T-1_2} - CT1_2 = 0,\\
&f_{n-1} : CT0_0 CT0_{n-7} + CT0_{n-1} + x_{(T-2)/2_1} + kr_{T-1_1} - CT1_1 = 0,\\
&f_n : CT0_{n-1}CT0_{n-8} + CT0_{n-2} + x_{(T-2)/2_0} + kr_{T-1_0} - CT1_0 = 0,\\
&f_{n+1} : x_{(T-2)/2_{n-2}}x_{(T-2)/2_{n-9}} + x_{(T-2)/2_{n-3}} + y_{(T-2)/2_{n-1}} + kr_{T-2_{n-1}} - CT0_{n-1} = 0,\\
&f_{n+2} : x_{(T-2)/2_{n-3}}x_{(T-2)/2_{n-10}} + x_{(T-2)/2_{n-4}} + y_{(T-2)/2_{n-2}} + kr_{T-2_{n-2}} - CT0_{n-2} = 0,\\
&\vdots\\
&f_{2n-2} : x_{(T-2)/2_1}x_{(T-2)/2_{n-6}} + x_{(T-2)/2_0} + y_{(T-2)/2_2} + kr_{T-2_2} - CT0_2 = 0,\\
&f_{2n-1} : x_{(T-2)/2_0}x_{(T-2)/2_{n-7}} + x_{(T-2)/2_{n-1}} + y_{(T-2)/2_1} + kr_{T-2_1} - CT0_1 = 0,\\
&f_{2n} : x_{(T-2)/2_{n-1}}x_{(T-2)/2_{n-8}} + x_{(T-2)/2_{n-2}} + y_{(T-2)/2_0} + kr_{T-2_0} - CT0_0 = 0,
\end{aligned}
\tag{15}
$$

Each pair of the remaining $(T-4)/2$ pairs of rounds is represented by a system of equations of the form (13), which consequently gives $(T-4)n$ equations of degree 2.

The number of additional binary variables required during the linearization process of the equations system representing the encryption algorithm is $(T-2)n$.

4.3. Generation of equations describing the round keys generation algorithm of the Simon cipher

The algorithm of generating round keys of the Simon$2n/mn$ cipher uses bitwise XOR operation and the right circular shift by 1 and 3 bits. The first m round keys are included in the main key and next kr_i keys, for $i = \overline{m, T-1}$ are generated with the use of the algorithm for generating round keys. All round keys are represented by binary variables, and no additional intermediate states are required. Hence, the number of variables needed to generate the system of polynomial equations describing the generation of round keys is $(T-1)n$. Equations are generated for each bit of the round key kr_i, for $i = \overline{m, T-1}$.

Let the n-bit round key kr_i be represented by a string of bits of the form:

$$kr_i = kr_{i_{n-1}}, \quad kr_{i_{n-2}}, \ldots, kr_{i_1}, \quad kr_{i_0},$$

where kr_{i_j} is the jth bit of the round key of round i. Right circular shift of the round key by 1 and 3 bits is performed, analogously to the encryption algorithm, as a cyclic bit shift of the string.

The given kr_i round key is determined from the equation (16) for $m = 2$ and $m = 3$ and from the equation (17) for $m = 4$.

$$kr_i = kr_{i-m} \oplus (kr_{i-1} \gg 3) \oplus (kr_{i-1} \gg 4) \oplus c \oplus (z_{\text{seq}})_{(i-m) \bmod 62}, \tag{16}$$

$$kr_i = kr_{i-m} \oplus kr_{i-3} \oplus (kr_{i-1} \gg 3) \oplus (kr_{i-1} \gg 4) \oplus (kr_{i-3} \gg 1) \oplus c \oplus (z_{\text{seq}})_{(i-m) \bmod 62}. \tag{17}$$

Based on the above equations, in which the operations are performed on n-bit words, a system of n polynomial equations over $GF(2)$ was generated for the round key kr_i. The obtained system was presented by the equation (18) for $m = 2$ and $m = 3$,

$$\begin{aligned}
&kr_{i-m_{n-1}} + kr_{i-1_2} + kr_{i-1_3} + c_{n-1} + (z_{\text{seq}})_{(i-m) \bmod 62_{n-1}} + kr_{i_{n-1}} = 0,\\
&kr_{i-m_{n-2}} + kr_{i-1_1} + kr_{i-1_2} + c_{n-2} + (z_{\text{seq}})_{(i-m) \bmod 62_{n-2}} + kr_{i_{n-2}} = 0,\\
&\vdots\\
&kr_{i-m_1} + kr_{i-1_4} + kr_{i-1_5} + c_1 + (z_{\text{seq}})_{(i-m) \bmod 62_1} + kr_{i_1} = 0,\\
&kr_{i-m_0} + kr_{i-1_3} + kr_{i-1_4} + c_0 + (z_{\text{seq}})_{(i-m) \bmod 62_0} + kr_{i_0} = 0
\end{aligned} \tag{18}$$

and by the equation (19) for $m = 4$

$$\begin{aligned}
&kr_{i-m_{n-1}} + kr_{i-3_{n-1}} + kr_{i-1_2} + kr_{i-1_3} + kr_{i-3_0} + c_{n-1} + (z_{\text{seq}})_{(i-m) \bmod 62_{n-1}} + kr_{i_{n-1}} = 0,\\
&kr_{i-m_{n-2}} + kr_{i-3_{n-2}} + kr_{i-1_1} + kr_{i-1_2} + kr_{i-3_{n-1}} + c_{n-2} + (z_{\text{seq}})_{(i-m) \bmod 62_{n-2}} + kr_{i_{n-2}} = 0,\\
&\vdots\\
&kr_{i-m_1} + kr_{i-3_1} + kr_{i-1_4} + kr_{i-1_5} + kr_{i-3_2} + c_1 + (z_{\text{seq}})_{(i-m) \bmod 62_1} + kr_{i_1} = 0,\\
&kr_{i-m_0} + kr_{i-3_0} + kr_{i-1_3} + kr_{i-1_4} + kr_{i-3_1} + c_0 + (z_{\text{seq}})_{(i-m) \bmod 62_0} + kr_{i_0} = 0.
\end{aligned} \tag{19}$$

All the generated equations are linear, and, therefore, no additional binary variables are required during linearization. On the other hand, the number of equations in the system representing the entire round key generation algorithm is $(T - m)n$.

4.4. Construction of an equations system representing the Simon cipher

Two approaches can be considered when an equations system describing the entire Simon cipher is constructed, depending on how the bits of the round keys are represented.

The round key bits are represented as binary variables in the first approach. Then the number of binary variables needed to generate the system of equations describing the T-round cipher Simon$2n/mn$ is equal to $mn+(T-m)n+(T-2)n$,

where:

- mn — number of binary variables for the key K,
- $(T-m)n$ — number of binary variables for the kr_i round keys,
- $(T-2)n$ — number of binary variables for the intermediate states x_i and y_i of the encryption algorithm, for one pair of plaintext - ciphertext.

The equations system describing the T-round Simon$2n/mn$ cipher consists of:

- n quadratic equations and n linear equations of the form (14),
- $(T-4)n$ quadratic equations of the form (13) and
- n linear equations and n quadratic equations of the form (15),

for the encryption algorithm and

- $(T-m)n$ linear equations, of the form (18) when $m=2$ or $m=3$ or (19) when $m=4$,

for the algorithm of generating round keys. During the linearization, $(T-2)n$ additional binary variables are required.

Let us consider the next step during the transformation of the equations system to the QUBO problem, i.e., the representation of the k_i multiplicity value by binary variables. This number is easy to determine because it depends on the number of monomials in the given equation for the equations generated over $GF(2)$. For the equations describing the encryption algorithm, each of the $(T-2)n$ quadratic equations consists of 5, and each of the $2n$ linear equations consists of 3 monomials. Hence, at most $(T-1)2n$, additional binary variables are required for these equations. In the case of the round key generation algorithm, each of the $(T-m)n$ linear equations consists of 5 monomials, for $m=2$ and $m=3$, or 7 monomials, for $m=4$. In both these cases, the number of additional binary variables is equal to $(T-m)2n$.

In a second approach, the round key bits are represented as polynomials that are XORed with the appropriate state in the encryption algorithm. The XOR operation is performed as an add, so it does not increase the degree of equations but increases the number of monomials. The number of binary variables needed to generate the equations system describing the entire cipher is $mn+(T-2)n$, where:

- mn — number of binary variables for the key K and
- $(T-2)n$ — number of binary variables for intermediate states of the encryption algorithm, for one pair of plaintext — ciphertext.

In this approach, in the equations, (16) and (17), the variables of the previous keys are substituted by the corresponding polynomials, resulting in the increasingly longer polynomials representing round keys for successive rounds.

The equations system describing the T-round Simon$2n/mn$ cipher consists of the equations (14), (13), and (15), where the kr_i denotes the appropriate polynomial. Since the bitwise XOR operation does not increase the degree of the equation, the number of binary variables necessary to perform the linearization operation is the same as in the previous approach. However, in this approach, the number of monomials in a given equation is not constant and it increases with the next round. For example, Table 1 shows the number of monomials and the corresponding number of binary variables needed for k_i multiples, for polynomials representing bits of round keys of successive rounds, for the Simon32/64 cipher. There are many different variants of the Simon cipher. Each variant has a different input block size and a different key length. Therefore, if the length of the input block is equal to the length of the key, then there is approximately one correct key for each pair of plaintext — ciphertext. Otherwise, if the length of the input block is less than the key length, for each such pair, there will be about 2^{mn-2n} keys appropriate, but only one key will be

TABLE 1. The number of monomials for polynomials representing bits of round keys of successive rounds for the Simon32/64 cipher.

Round	Number of monomials	Number of variables	Round	Number of monomials	Number of variables
0	1	0	16	29 or 30	4
1	1	0	17	33 or 34	5
2	1	0	18	33 or 34	5
3	1	0	19	29 or 30	4
4	5 or 6	2	20	29 or 30	4
5	9 or 10	3	21	27 or 28	4
6	17 or 18	4	22	25 or 26	4
7	19 or 20	4	23	25 or 26	4
8	21 or 22	4	24	29 or 30	4
9	23 or 24	4	25	27 or 28	4
10	27 or 28	4	26	31 or 32	4 or 5
11	29 or 30	4	27	29 or 30	4
12	27 or 28	4	28	29 or 30	4
13	27 or 28	4	29	27 or 28	4
14	27 or 28	4	30	33 or 34	5
15	29 or 30	4	31	29 or 30	4

the correct one if the other pairs are considered. In order to find the right key with high probability, $\lceil \frac{m}{2} \rceil$ pairs of plaintext — ciphertext is required. Therefore, for variants for which the input block length is less than the key length, the encryption algorithm equations were generated for two different plaintext — ciphertext pairs, creating two equations systems describing the encryption algorithm.

For each instance of the Simon$2n/mn$ cipher, the size of the QUBO problem was determined for both approaches. The obtained results are presented in Table 2, where for a given instance, the top row contains the results for the first approach, and the bottom row contains the results for the second approach. As the obtained results show, when the key length is equal to the input block length, the second approach is more efficient because the size of the obtained QUBO problem is smaller than for the first approach. For the remaining variants, constructing a system of equations representing the cipher according to the first approach is more efficient.

TABLE 2. The QUBO problem size for variants of Simon$2n/mn$ cipher.

Simon $2n/mn$	Rounds	Pairs	Equations	Initial number of variables	Linear k_i	Size of multiples	QUBO
32/64	32	2	1,472	1,472	960	2,880	5,312
			1,024	1,024	960	4,240	6,224
48/72	36	2	2,520	2,496	1,632	4,944	9,072
			1,728	1,704	1,632	6,538	9,874
48/96	36	2	2,496	2,496	1,632	4,896	9,024
			1,728	1,728	1,632	7,680	11,040
64/96	42	2	3,936	3,904	2,560	7,744	14,208
			2,688	2,656	2,560	11,100	16,316
64/128	44	2	4,096	4,096	2,688	8,064	14,848
			2,816	2,816	2,688	13,308	18,812
96/96	52	1	4,896	4,896	2,400	9,696	16,992
			2,496	2,496	2,400	9,966	14,862
96/144	54	2	7,632	7,584	4,992	15,072	27,648
			5,184	5,136	4,992	22,998	33,126
128/128	68	1	8,576	8,576	4,224	17,024	29,824
			4,352	4,352	4,224	18,694	27,270
128/256	72	2	13,568	13,568	8,960	26,880	49,408
			9,216	9,216	8,960	51,128	69,304

5. Algebraic attack with the use of quantum annealing on the Simon cipher

It is difficult to speculate whether the proposed attack might be better than a full search or a Grover attack. The computational complexity of solving the QUBO problem by quantum annealing still requires much research. Using the heuristics presented in [9], it can be estimated that the expected time of solving a QUBO problem consisting of N binary variables is equal to $O(e^{\sqrt{N}})$. Assuming this complexity is true, we estimated the number of rounds for which the proposed attack is better than the full search attack. The results obtained for each variant of the Simon$2n/mn$ cipher are presented in Table 3, where in parentheses is the percentage of the attacked number of rounds is shown. Unfortunately, neither version has achieved the result better than the best classic attack.

In previous papers, the application of the proposed attack to an AES cipher in [5] and a Speck cipher in [4] was presented. Table 5 compares the obtained sizes of QUBO problems for these ciphers. The size of the QUBO problem for the Simon cipher, with a key length of 128 bits, is 90 % of the size of the QUBO problem for the AES cipher, with the same key size, and 140 % of the size of the QUBO problem of the Speck cipher. However, for variants with a key length of 256 bits, the Simon cipher's QUBO problem size is 68 % of the AES cipher's QUBO problem size and 145 % of the Speck cipher's QUBO problem size.

TABLE 3. The number of rounds for which the proposed attack is better than a full search for each variant of the Simon $2n/mn$ cipher.

Simon$2n/mn$	Number of rounds	The best classic attack	Our attack
Simon32/64	32	24 (75 %)	13 (41 %)
Simon48/72	36	24 (67 %)	11 (31 %)
Simon48/96	36	25 (69 %)	18 (50 %)
Simon64/96	42	30 (71 %)	14 (33 %)
Simon64/128	44	31 (70 %)	24 (55 %)
Simon96/96	52	37 (71 %)	16 (31 %)
Simon96/144	54	38 (70 %)	20 (37 %)
Simon128/128	68	49 (72 %)	22 (32 %)
Simon128/256	72	53 (74 %)	46 (64 %)

In order to verify the correctness of the generated equations systems, an attack was performed on a smaller instance of the Simon cipher, using the D-Wave quantum annealer. A smaller instance of the Simon$2n/mn$ cipher was designed with the following parameters:

- block length: $2n = 16$ bits,
- word length: $n = 8$ bits,
- key length: $mn = 16$ bits,
- number of keywords: $m = 2$,
- bit sequence for generation of round keys: z_0,
- the number of rounds: $T = 6$.

The value of the penalty polynomial coefficient was set to $\mathrm{Pen}_{\mathrm{lin}} = 6$. As a result of the transformation, the QUBO problem consisting of 196 binary variables was obtained. The Matrix Q of the generated problem is shown in Fig. 8, where the blue points denote the non-zero coefficients of the QUBO polynomial problem. The absolute values of the coefficients of the obtained Q matrix are in the range from 2 to 29, which, after scaling in the D-Wave quantum annealer, took values from 0.069 to 1. The number of non-zero elements in the above, upper-triangular Q matrix is $1,684$ out of $19,306$ of all possible elements, which is $8.72\,\%$. The annealing time of the QUBO problem of the Simon16/16 cipher was set to 20 minutes, after which a solution containing the correct key was obtained.

TABLE 4. Comparison of the size of QUBO problems for Simon, Speck, and AES ciphers.

Cipher variant	Number of rounds	Size of QUBO problem
Simon128/128	68	$27,270$
Speck128/128	32	$19,311$
AES128	10	$29,770$
Simon128/256	72	$49,408$
Speck128/256	34	$33,721$
AES256	14	$72,597$

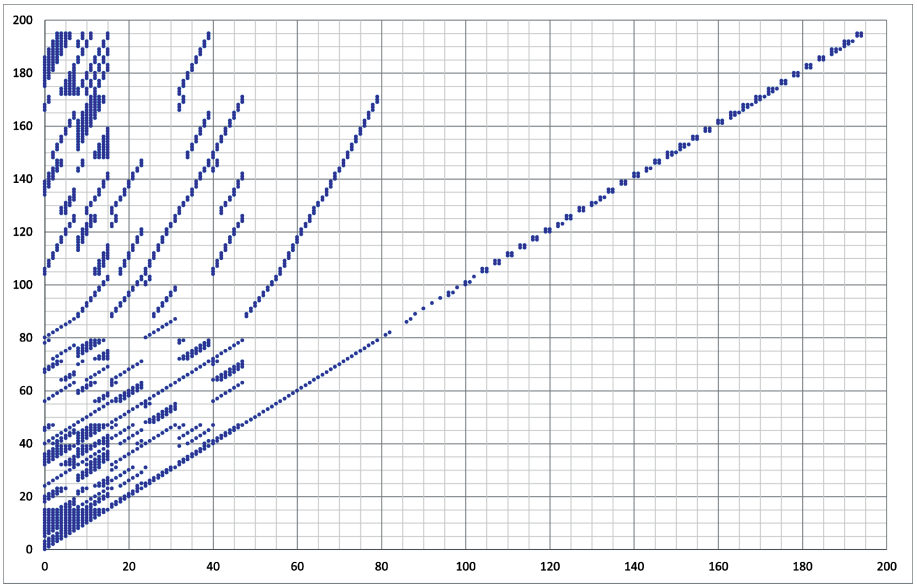

FIGURE 8. The Q matrix of Simon16/16 QUBO problem.

TABLE 5. Comparison of the size of QUBO problems for Simon, Speck, and AES ciphers.

Cipher variant	Number of rounds	Size of QUBO problem
Simon128/128	68	27,270
Speck128/128	32	19,311
AES128	10	29,770
Simon128/256	72	49,408
Speck128/256	34	33,721
AES256	14	72,597

For the comparison of the impact of the density of the Q matrix on the efficiency of solving the QUBO problem using the D-Wave computer, a small instance of the Speck cipher was designed, so that the obtained size of the QUBO problem was approximately equal to the size of the QUBO problem of the Simon16/16 cipher.

The following parameters were assumed for the Speck cipher:

- block length: $2n = 16$ bits,
- word length: $n = 8$ bits,
- key length: $mn = 16$ bits,
- number of keywords: $m = 2$,
- number of bits for right rotation: $\alpha = 4$,
- number of bits for left rotation: $\beta = 2$,
- the number of rounds: $T = 3$.

The equations representing the Speck16/16 cipher were generated over $GF(2)^8$. The penalty coefficient was assumed to be $Pen_{\text{lin}} = 6$. The obtained QUBO problem consists of 170 binary variables. The matrix Q of the generated problem is shown in Fig. 9. This matrix contains values of the coefficients from 1 to $14,680,064$, which means that after scaling in the annealer, the values of the coefficients range from $6.8{\cdot}10^{-8}$ to 1. Additionally, the number of all elements in the upper-triangular matrix Q is $14,365$, out of which $4,902$ are non-zero, is $34\,\%$. Consequently, this matrix is denser than the matrix of the solved QUBO problem of the Simon16/16 cipher. The expected minimum energy value is $-38,203$. The annealing time was set to 20 minutes. The obtained minimum energy is equal to $-37,530$, and the obtained annealing result does not agree with the key used.

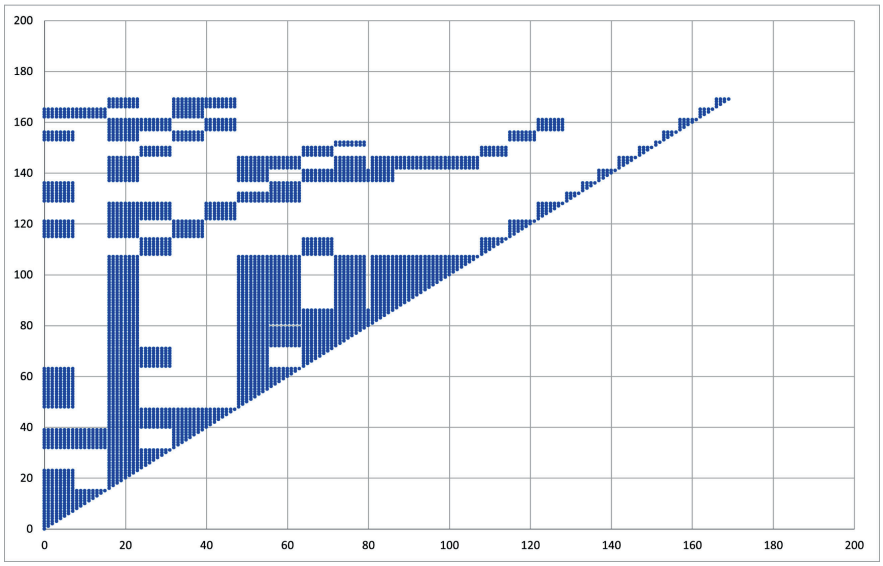

FIGURE 9. The Q matrix of Speck16/16 QUBO problem.

Since the Q matrix of the QUBO problem for the Speck16/16 cipher turned out to be denser than for the Simon16/16 cipher, and the scaled coefficients turned out to be very small, the QUBO problem was generated for the Speck cipher, described by equations over $GF(2)$. In order to obtain a comparable size of the QUBO problem, the following parameters were assumed:

- block length: $2n = 8$ bits,
- word length: $n = 4$ bits,
- key length: $mn = 8$ bits,
- number of keywords: $m = 2$,
- number of bits for right rotation: $\alpha = 3$,
- number of bits for left rotation: $\beta = 2$,
- the number of rounds: $T = 3$.

After transforming the equations system over $GF(2)$ into the QUBO problem, the problem of the size of 174 binary variables was obtained. The matrix Q of the obtained problem is shown in Fig. 10. It consists of $1,023$ non-zero coefficients, which is $6.7\,\%$ of all elements of the Q matrix. The values of the obtained coefficients range from 1 to 39, which after scaling gives the range from 0.026 to 1. This problem was solved within 20 minutes by a D-Wave quantum annealer, and the correct key was obtained.

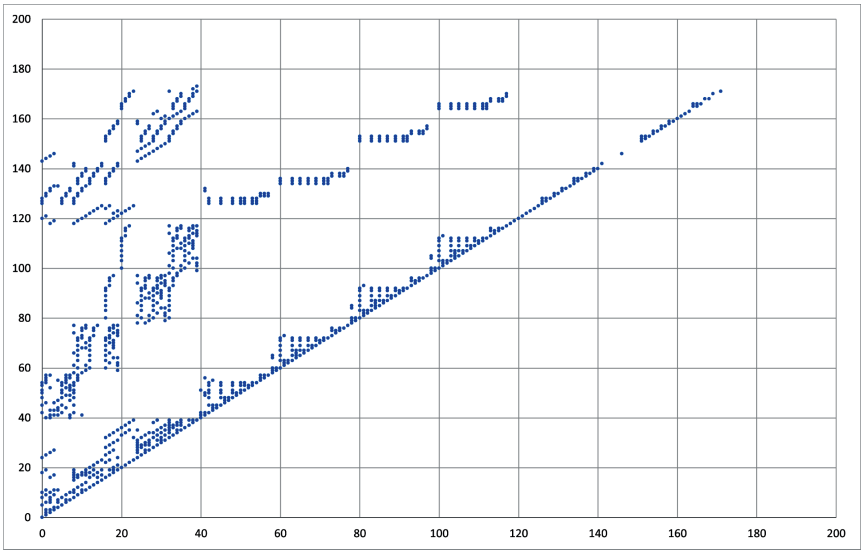

FIGURE 10. The Q matrix of Speck8/8 QUBO problem.

6. Conclusion

This article presents the application of an algebraic attack using quantum annealing to the Simon cipher having a Feistel structure. We have shown how to get a QUBO problem of the smallest possible size. The article also demonstrates the fact that the size of the QUBO problem is not a sufficient factor in the effectiveness of solving the problem with the D-Wave quantum computer. An equally important factor is the structure and density of the Q matrix of the QUBO problem, which practical experiments have confirmed. The experiments have also shown that a given problem could be transformed in various ways, some of which may be more effective than others. If the resulting problem does not provide a correct solution, one could consider alternative formulas. For example, a transformation that increases the number of variables in the target QUBO problem may be more effective for embedding it in a hardware graph and when looking for a solution.

Moreover, the computational complexity of solving Ising or QUBO problems with a D-Wave computer has still not been fully studied. However, it can be assumed that if the Q matrix of the Simon128/128 cipher's QUBO problem is at most as dense as the Q matrix of the AES128 cipher's QUBO problem, then a smaller number of variables will cause that the problem will be solved in a shorter time. Unfortunately, the size of the current quantum annealing resources does not allow any variant of the Simon cipher to be broken in practice.

REFERENCES

[1] *D-Wave System Documentation*. Available at: `https://docs.dwavesys.com/docs/latest/index.html`, access: 2022-06-10.

[2] BEAULIEU, R.—SHORS, D.—SMITH, J.—TREATMAN-CLARK, S.—WEEKS, B.—WINGERS, L.: *The SIMON and SPECK families of lightweight block ciphers*, Cryptology eprint archive (2013).

[3] BORLE, A.—LOMONACO, S. J.: *Analyzing the quantum annealing approach for solving linear least squares problems*. In: *International Workshop on Algorithms and Computation*, Lecture Notes in Comput. Sci., Vol. 11355, Springer, Cham, 2019, pp. 289–301.

[4] BUREK, E.—WROŃSKI, M.: *Quantum Annealing and Algebraic Attack on Speck Cipher*. In: *Computational Science — ICCS 2022* (D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, P. M. A. Sloot, eds.), Springer International Publishing. Cham. 2022, pp. 143–149,

[5] BUREK, E.—WROŃSKI, M.—MAŃK, K.—MISZTAL, M.: *Algebraic attacks on block ciphers using quantum annealing*, IEEE Transactions on Emerging Topics in Computing **10** (2022), 678–689.

[6] CHEN, H.—WANG, X.: *Improved linear Hull attack on round-reduced Simon with dynamic key-guessing techniques*. In: International Conference on Fast Software Encryption, Springer, Cham, 2016, pp. 428–449.
file:///C:/Users/ivana/Downloads/978-3-662-52993-5.pdf

[7] CHU, Z.—CHEN, H.—WANG, X.—DONG, X.—LI, L.: *Improved integral attacks on SIMON32 and SIMON48 with dynamic key-guessing techniques*, Security and Communication Networks **2018**, no. 1, (2018). https://doi.org/10.1155/2018/5160237

[8] JIANG, S.—BRITT, K. A.—MCCASKEY, A. J.—HUMBLE, T. S.—KAIS, S.: *Quantum annealing for prime factorization*, Scientific reports **8** (2018), 1–9.

[9] MUKHERJEE, S.—CHAKRABARTI, B. K.: *Multivariable optimization: Quantum annealing and computation*, The European Physical Journal Special Topics **224** (2015), 17–24.

[10] RADDUM, H.: *Algebraic analysis of the simon block cipher family*. In: *Progress in Cryptology–LATINCRYPT 2015*: 4th International Conference on Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23-26, 2015, Proceedings 4., Springer International Publishing, 2015, pp. 157–169.

[11] ROSENBERG, I. G.: *Reduction of bivalent maximization to the quadratic case*. Cahiers Centre Études Rech. Opér. **17** (1975), 71–74.

[12] SHOR, P. W.: *Algorithms for quantum computation: discrete logarithms and factoring*. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science,(Santa Fe, NM, 1994)*, IEEE Comput. Soc. Press, Los Alamitos, CA, 1994. pp. 124–134.

[13] WRONSKI, M.: *Solving discrete logarithm problem over prime fields using quantum annealing and n^32 logical qubits*, IACR Cryptol. ePrint Arch. **2021** (2021), Paper no. 2021/527. https://eprint.iacr.org/2021/527

Received September 30, 2022
Revised October 10 , 2024
Accepted January 20, 2024
Publ. online June 27, 2025

Cybernetics Faculty
Military University of Technology
Kaliskiego 2
00-908 Warsaw
POLAND
E-mail: elzbieta.burek@wat.edu.pl