Quantum annealing and algebraic attack on Speck cipher*

Elżbieta Burek[1][0000−0003−2937−0833], Michał Wroński[1][0000−0002−8679−9399]

Military University of Technology, Kaliskiego Str. 2, Warsaw, Poland
{elzbieta.burek, michal.wronski}@wat.edu.pl

Abstract. Algebraic attacks using quantum annealing are a new idea of cryptanalysis. This paper shows how to obtain a QUBO problem equivalent to the algebraic attack on the Speck cipher, using as small a number of logical variables as possible. The main idea of minimizing the number of variables in the algebraic attack on this ARX cipher was appropriate cipher partition and insertion of additional variables. Using such an idea, in the case of the most popular variants: Speck-128/128 and Speck-128/256, the equivalent QUBO problem has 19,311 and 33,721 logical variables, which is more efficient than the same attack on AES cipher, where for AES-128 and AES-256, an equivalent QUBO problem consist of 29,770 and 72,597 logical variables, respectively. It is an open question if this kind of attack may overtake, in some cases, brutal or Grover's attack.

Keywords: Cryptanalysis, algebraic attacks, Speck, D-Wave Advantage, quantum annealing

1 Introduction

Quantum computing has allowed the development of new approaches to computational problems that classical computers cannot cope with. One such problem in cryptanalysis of block ciphers is solving large systems of multivariate polynomial equations during algebraic attacks. In general, the idea of algebraic attacks is based on two steps: the first is to represent the cipher as a system of multivariate polynomial equations, and the second is to solve the created system.

In [3] Burek et al. showed how to transform obtained system of multivariate equations into the QUBO problem.

QUBO (Quadratic Unconstrained Binary Optimization) is a combinatorial optimization problem in which the cost function $f(x)$ is defined on an n-dimensional binary vector space $\mathbb{B}^n$ onto $\mathbb{R}$, as follows: $QUBO: \ min\ f(x) = x^tQx$, where x is a vector of binary variables and Q is an upper diagonal matrix of real weights.

* This work was supported by the Military University of Technology's University Research Grant No. 858/2021: "Mathematical methods and models in computer science and physics."

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

Since the variables are binary, $x_i^2 = x_i$ holds, and the cost function can be represented as: $QUBO: \ min\ f(x) = \sum_i Q_{i,i}x_i \ + \sum_{i<j} Q_{i,j}x_i x_j$.

It is worth noting that algebraic attacks on symmetric ciphers using general-purpose quantum computing have been studied in [4], [6], where variants of the HHL [7] algorithm has been used.

The contribution presented in this paper is the presentation of the application of an algebraic attack using quantum annealing on the Speck cipher. We focused on obtaining equivalent QUBO problem using as small variables as possible. The main idea of minimizing the number of variables in the algebraic attack on the Speck cipher was appropriate ciphers partition and insertion of additional variables. In the case of the most popular variants: Speck128/128 and Speck128/256, we obtained the equivalent QUBO problem consisting of 19,311 and 33,721 logical variables. According to our experiments, applying quantum annealing to the algebraic attacks on Speck should be much more efficient than the same attack on AES cipher, where in the case of the algebraic attack on AES-128 and AES-256, an equivalent QUBO problem consist of 29,770 and 72,597 logical variables respectively. It is an open question if this kind of attack may overtake, in some cases, brutal or Grover's attacks. However, assuming that complexity of solving of QUBO problem consisting of N variables requires $O\left(e^{\sqrt{N}}\right)$ elementary operations [8], one can obtain an attack faster than the brute force on Speck-128/256 consisting of 31 of 34 rounds, which is better than the best known classical attack on this cipher variant, which works for 25 rounds.

2 Algebraic attack on Speck using Quantum annealing

This section will present the method of representing the Speck cipher using multivariate polynomial equations to obtain a system of multivariate polynomial equations with as few monomials as possible, which consequently allows obtaining a problem in the QUBO form with as few binary variables as possible.

2.1 Speck cipher

The Speck cipher is a family of lightweight block ciphers of the ARX type, presented in [2] as highly-optimized block ciphers intended for software and hardware implementations.

An instance of the Speck cipher will be designated, according to [2], as Speck$2n/mn$, where $2n$ is the length of the input block, n is the word length, and mn is the key length. The Speck$2n/mn$ cipher uses the n-bit word operations, as bitwise xor, addition modulo 2^n and right and left rotations.

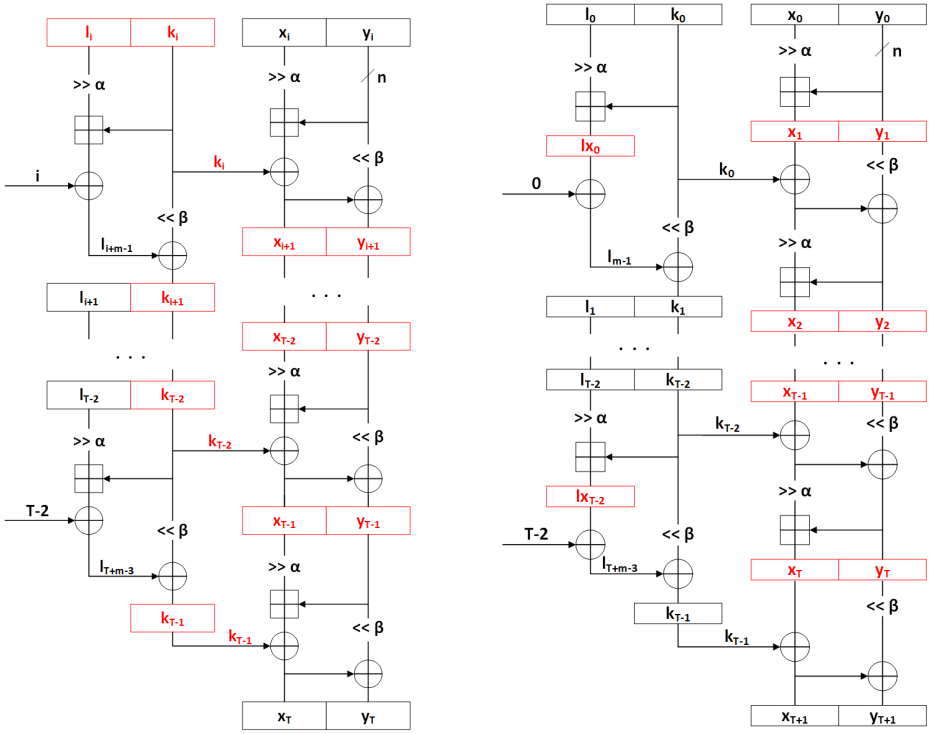
The general structure of the Speck$2n/mn$ cipher is shown in Figure 1, where T denotes the number of rounds, $\oplus$ denotes the bitwise xor operation, $\boxplus$ denotes the addition modulo 2^n, and $\gg \alpha$ and $\ll \beta$ denote a right rotation by α and left rotation by β bits, respectively.

The round function of the encryption algorithm of the Speck$2n/mn$ cipher is a map $R : GF(2)^n \times GF(2)^n \to GF(2)^n \times GF(2)^n$, where $GF(q)$ is Galois field

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

with q elements, defined as follows: $R(x_{i+1}, y_{i+1}) = (((x_i \gg \alpha) + y_i) \oplus k_i, (y_i \ll \beta) \oplus ((x_i \gg \alpha) + y_i) \oplus k_i)$, where x_i and y_i is, respectively, the left and the right n-bit word of the input block of i round, k_i is the round key and i is the number of the round.

The round key generation algorithm uses the round function. The key is divided into m n-bit words, where the least significant n bits are the round key of the first round, and the next n-bit words are successive l_i words: $K = [l_{m-2}, \ldots, l_0, k_0]$, where $l_i, k_0 \in GF(2)^n$. The words l_i and the round keys k_i are determined as: $l_{i+m-1} = ((l_i \gg \alpha) + k_i) \oplus i$ and $k_{i+1} = (k_i \ll \beta) \oplus l_{i+m-1}$.



(a) Split of Speck2n/mn cipher according to cipher documentation.

(b) Split of Speck2n/mn cipher by using additional variables.

Fig. 1: Structure of the Speck2n/mn cipher for the presented approaches.

2.2 Efficient approach to generating multivariate polynomial equations

In the approach to generating multivariate polynomial equations, where the range of the round was held by the Speck algorithm documentation, the ad-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

ditional binary variables have been introduced for intermediate states between rounds and round keys. The number of additional binary variables for intermediate states is $(T-1)2n$, and for round keys $(T-1)n$, as presented in red in Figure 1a. The multivariate polynomials were generated over $GF(2)^n$, separately for the left and right words, each state between rounds, and each round key. Finally, the degree of the left word polynomial equations is $2n+1$, so the number of binary variables in the QUBO problem will be very large.

In our approach to generating multivariate polynomial equations, the range of the round was changed. Figure 1b shows in red how the additional intermediate variables were introduced. Additional binary variables were introduced for round keys and intermediate states, which were introduced after the addition modulo 2^n in the encryption algorithm and the round key generation algorithm. Since there is no key addition in the first round in Figure 1b, the bits of the words x_1 and y_1 are also known: $x_1 = (x_0 \gg \alpha) + y_0$ and $y_1 = y_0$.

The number of additional binary variables for the round keys is $(T-1)n$, for intermediate states in the encryption algorithm, it is $(T-1)2n$, and for intermediate states lx_i, in the round key generation algorithm, it is $2n$.

The xor operation of a_j and b_j bits may be written as $a_j \oplus b_j = a_j + b_j - 2a_jb_j$, therefore for the n-bit a and b words is executed as: $a \oplus b = \sum_{j=0}^{n-1} 2^j(a_j + b_j - 2a_jb_j)$. Since in this approach addition modulo 2^n is executed after the xor operation, then: $(a+b)\ mod\ 2^n = \sum_{j=0}^{n-1} 2^j(a_j + b_j) - c \cdot 2^n$, where the bit c is the carry bit of sum.

In this approach, the equation representing the left-word of one round of the encryption algorithm, except the last round, takes the following form: $x_{i+1} = (((x_i \oplus k_i) \gg \alpha) + y_{i+1})\ mod\ 2^n$, and after performing all operations it is form as:

$$\begin{aligned}&\sum_{j=0}^{n-1} 2^j((x_i)_{(j+\alpha)mod\ n} + (k_i)_{(j+\alpha)mod\ n} - 2(x_i)_{(j+\alpha)mod\ n}(k_i)_{(j+\alpha)mod\ n} + \\ &+ (y_{i+1})_j - (x_{i+1})_j) - c \cdot 2^n = 0.\end{aligned} \tag{1}$$

The equation representing the right-word of one round of the encryption algorithm, except the last round, takes the following form: $y_{i+1} = ((y_i \ll \beta) \oplus x_i \oplus k_i)$, which can be finally converted to:

$$\begin{aligned}&\sum_{j=0}^{n-1} 2^j((y_i)_{(j-\beta)mod\ n} + (x_i)_j - 2(y_i)_{(j-\beta)mod\ n}(x_i)_j + (k_i)_j + \\ &- 2(k_i)_j(y_i)_{(j-\beta)mod\ n} - 2(k_i)_j(x_i)_j + 4(k_i)_j(y_i)_{(j-\beta)mod\ n}(x_i)_j - (y_{i+1})_j) = 0.\end{aligned} \tag{2}$$

Similarly, the last round of the encryption algorithm can be represented by the following equations: $x_{T+1} = (x_T \oplus k_{T-1})$, for the left word, which is equivalent to:

$$\sum_{j=0}^{n-1} 2^j((x_T)_j + (k_{T-1})_j - 2(x_T)_j(k_{T-1})_j - (x_{T+1})_j) = 0, \tag{3}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

and for the right word: $y_{T+1} = (y_T \ll \beta) \oplus x_{T+1}$, which is equivalent to:

$$\sum_{j=0}^{n-1} 2^j((y_T)_{(j-\beta) mod\ n} + (x_{T+1})_j - 2(y_T)_{(j-\beta) mod\ n}(x_{T+1})_j - (y_{T+1})_j) = 0. \quad (4)$$

Two multivariate polynomial equations also represent each round of the round key generation algorithm. The first equation associates the binary variables of the word l_i with the binary variables of the word lx_i, and the second equation relates the binary variables of the word lx_i with the binary variables of the k_i and k_{i+1} round keys.

The equation defining the lx_i word is as follows: $lx_i = ((l_i \gg \alpha) + k_i)\ mod\ 2^n$, which can be converted to the form:

$$\sum_{j=0}^{n-1} 2^j((l_i)_{(j+\alpha) mod\ n} + (k_i)_j - (lx_i)_j) - c \cdot 2^n = 0. \quad (5)$$

The equation defining the k_{i+1} round key has the following form: $k_{i+1} = ((lx_i \oplus i) \oplus (k_i \ll \beta))$, which is equivalent to:

$$\begin{aligned} &\sum_{j=0}^{n-1} 2^j((lx_i)_j + (i)_j - 2(lx_i)_j(i)_j + (k_i)_{(j-\beta) mod\ n} - 2(lx_i)_j(k_i)_{(j-\beta) mod\ n} + \\ &- 2(i)_j(k_i)_{(j-\beta) mod\ n} + 4(lx_i)_j(i)_j(k_i)_{(j-\beta) mod\ n}) = 0, \end{aligned} \quad (6)$$

where $(i)_j$ is the j-th bit of known constant i. The degree of the polynomial in Equation (6) is 3. However, such a degree occurs only in the monomial with the i constant bit, so the monomial will have the degree 2 if the constant bit is 1, otherwise, the monomial will vanish.

In the proposed approach to generating multivariate polynomial equations representing the Speck$2n/mn$ cipher, the degree of polynomials is constant and does not depend on the length of the input block. T-round Speck$2n/mn$ cipher can be represented by the system of: $T-1$ polynomials of degree 2, of the form as in Equation (1), $T-1$ polynomials of degree 3, of the form as in Equation (2), one polynomial of degree 2, of the form as in Equation (3) and one polynomial of degree 2, of the form as in Equation (4) for the encryption algorithm. Additional, for the round key generation algorithm: $T-1$ polynomials of degree 1, of the form as in Equation (5) and $T-1$ polynomials of degree 2, of the form as in Equation (6).

3 Transformation of algebraic attacks on Speck using quantum annealing

Cryptanalysis of Speck algorithm has been widely described, see [1], [10], [5]. This section will describe the results of the transformation of algebraic attacks using quantum annealing on Speck.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

We used the transformation method of algebraic attacks to the QUBO problem presented in [3].

It is worth presenting the following observation, which can be found in [9]. First, let us note that there are many different variants of the Speck cipher. Each variant has a different block size ($2n$) and key length (mn). If $2n \geq mn$, there is approximately one proper key for each pair of plaintext - ciphertext. Things are getting different if $2n < mn$. In such a case, having only one pair of plaintext - ciphertext, for each pair, approximately 2^{mn-2n} keys will be proper for this pair, but only one will be proper for all other pairs. If one wants to find the key used for encryption with high probability, in such a case, there are required $\lceil \frac{m}{2} \rceil$ plaintext - ciphertext pairs.

For each variant of the Speck cipher, we computed the number of variables of the equivalent QUBO problem. Let us note that for variants in which block length is smaller than key length, we used 2 pairs of plaintext - ciphertext, and therefore, in such cases, such QUBO problem is constructed from two smaller QUBO problems - each problem for each pair. Unfortunately, our final QUBO problem must consist of two smaller. We most frequently obtain some proper solution using quantum annealing, but not all solutions. It means that we cannot solve such systems independently for each pair.

Table 1: Results of transformation of the system of multivariate quadratic equations describing the AES and Speck ciphers to the QUBO problem.

Cipher variant	Number of rounds	Number of variables	Cipher variant	Number of rounds	Number of variables
AES-128	10	29,770	Speck96/96	28	12,418
AES-192	12	62,153	Speck64/128	27	13,711
AES-256	14	72597	Speck128/128	32	19,311
Speck32/64	22	5,789	Speck96/144	29	21,716
Speck48/72	22	8,470	Speck128/192	33	32,659
Speck48/96	23	8,884	Speck128/256	34	33,721
Speck64/96	26	13,167			

Because QUBO problem, equivalent to the algebraic attack on Speck cipher, in general, consists of less number of variables than analogic QUBO problem in the case of AES cipher with the same block size and key size (see Table 1), we conclude, that Speck cipher is easier to break using quantum annealing. However, it is hard to speculate if this attack can outperform brute force or Grover's attack. The computational complexity of solving the QUBO problem using quantum annealing still requires much research. However, it is claimed that such complexity depends mostly on the number of variables. The precise time complexity of solving the QUBO problem using quantum annealing has not been computed yet. Using heuristics, it is possible to estimate the expected time of solving QUBO problem consisting of N variables as $O\left(e^{\sqrt{N}}\right)$ [8]. Unfortunately, using current quantum annealers, it is impossible to break any variant of Speck cipher in practice.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12

4 Conclusion

This paper presents the transformation of the algebraic attack on the Speck cipher to the QUBO problem. We showed how to obtain the smallest possible number of variables for a QUBO problem. To obtain such a small number of variables, we proposed a novel way of describing the algebraic structure of each of the algorithms.

The computational complexity of solving the QUBO problem using quantum annealing has not been fully studied yet, and much more research in this area is required.

Further works should be more research on the computational complexity of solving algebraic attacks on cryptographic algorithms using quantum annealing and applying the presented method to other symmetric algorithms.

References

1. Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. Differential cryptanalysis of round-reduced simon and speck. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption*, pages 525–545, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
2. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *cryptology eprint archive*, 2013.
3. Elbieta Burek, Micha Wroski, Krzysztof Mak, and Micha Misztal. Algebraic attacks on block ciphers using quantum annealing. *IEEE Transactions on Emerging Topics in Computing*, In press.
4. Yu-Ao Chen and Xiao-Shan Gao. Quantum algorithm for boolean equation solving and quantum algebraic attack on cryptosystems. *Journal of Systems Science and Complexity*, pages 1–40, 2021.
5. Ashutosh Dhar Dwivedi, Pawel Morawiecki, and Gautam Srivastava. Differential cryptanalysis of round-reduced speck suitable for internet of things devices. *IEEE Access*, 7:16476–16486, 2019.
6. Juntao Gao, Hao Li, Baocang Wang, and Xuelian Li. Quantum security of aes-128 under hhl algorithm. *Quantum Information and Computation*, 22(3&4):0209–0240, 2022.
7. Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
8. Sudip Mukherjee and Bikas K Chakrabarti. Multivariable optimization: Quantum annealing and computation. *The European Physical Journal Special Topics*, 224(1):17–24, 2015.
9. Aleksey I Pakhomchik, Vladimir V Voloshinov, Valerii M Vinokur, and Gordey B Lesovik. Converting of boolean expression to linear equations, inequalities and qubo penalties for cryptanalysis. *Algorithms*, 15(2):33, 2022.
10. Ling Song, Zhangjie Huang, and Qianqian Yang. Automatic differential analysis of arx block ciphers with application to speck and lea. In *Australasian Conference on Information Security and Privacy*, pages 379–394. Springer, 2016.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_12