Unsafe mechanisms of Bluetooth, E_0 stream cipher cryptanalysis with quantum annealing

Mateusz Leśniak[1][0009−0001−0092−2975], Elżbieta Burek[2][0000−0003−2937−0833], and Michał Wroński[1][0000−0002−8679−9399]

[1] NASK National Research Institute, Kolska Str. 12, Warsaw, Poland
{mateusz.lesniak, michal.wronski}@nask.pl
[2] Military University of Technology, Kaliskiego Str. 2, Warsaw, Poland
elzbieta.burek@wat.edu.pl

Abstract. Due to Shor's and Grover's algorithms, quantum computing has become one of the fastest-evolving areas in computational science. Nowadays, one of the most interesting branches of quantum computing is quantum annealing. This paper presents the efficient method of transforming stream cipher E0 to the QUBO problem and then retrieving the Encryption Key of this cipher using quantum annealing. Comparably to other asymmetric and symmetric cryptographic algorithms, the presented transformation is efficient, requiring only 2,728 (2,751) logical variables for attack with 128 (129) consecutive keystream bits. According to our knowledge, it is the most efficient algorithm transformation with a 128-bit key. Moreover, we show that using current quantum annealers, one can embed the attack for E0 for 58 consecutive bits of keystream, from 128 (129), which are necessary for the attack's first stage (second stage). Therefore, it is likely that it will be possible to embed E0 on available quantum annealers in the next few years.

Keywords: Stream cipher · Bluetooth · E_0 cipher · Cryptanalysis · Quantum annealing

1 Introduction

Quantum computing is an approach that evolves fast nowadays. The (r)evolution in this aspect touches both theoretical and practical aspects, such as building bigger and bigger quantum computers. For cryptological society, several aspects are the most important nowadays. One is: How large are cryptographic problems we can solve using available quantum computers nowadays? This question is also important as quantum annealers are also special cases of quantum computers.

The application of quantum annealing to cryptanalysis of stream ciphers was first presented in [13], in the context of Grain 128 and Grain 128a ciphers. The efficient transformation of algebraic attacks on these two ciphers to the QUBO problem required 5,751 and 6,761 logical variables, respectively. It means that the algebraic attack using quantum annealing is a serious threat because there is a large probability that in several years, it will be possible to build a dedicated

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

quantum annealer on which it will be able to run such algebraic attacks on some stream ciphers.

This paper presents another step in applying quantum annealing to the cryptanalysis of stream ciphers.

The main differences between this paper and [13] are:

1. E0 stream cipher is practically used, for example, in Bluetooth communication protocol, while it is hard to find any practical applications of the Grain cipher.
2. In some cases, which will be described in detail later, it is possible to obtain the whole Encryption key of the E0 cipher running transformation to the QUBO problem twice. Because the same Encryption key is used for 23,5 hours, having only 129 consecutive bits of stream, one can retrieve almost the whole 1-day communication using E0 cipher with 50% probability.
3. According to our research and simulations, the creation of a dedicated quantum annealer on which it would be possible to run an algebraic attack on a full E0 cipher should be much easier and more probable than in the case of Grain ciphers family because our attack might be run on the dedicated quantum annealer with 2,751 qubits and 18,156 couplers, for both obtained QUBO problems. On the other hand, the dedicated quantum annealer for Grain 128 (Grain 128a) requires 5,751 (6,761) qubits and 77,496 (94,865) couplers.
4. The QUBO problem for an algebraic attack on the E0 cipher with a 128-bit keystream is too large to embed even in the latest Zephyr architecture. However, smaller problems were generated for the unchanged E0 algorithm but for a shorter keystream. The biggest problem embedded in the Zephyr architecture was generated for a 58-bit keystream.

2 Bluetooth encryption overview

Communication over short distances, provided by Bluetooth, protects confidentiality. The security mechanism has evolved over the versions of Bluetooth standards. The following paper focuses on the Legacy mechanism. As presented in [3], Legacy encryption is performed if at least one device does not support Secure Connections and its features. Legacy encryption is performed with the E0 stream cipher, derived from the Massey-Rueppel algorithm.
Before encryption starts, the connection between devices must be established. Each device should carry an initialization phase in order to generate and exchange link keys. The initialization phase consists of five steps:

- generation of an initialization key;
- generation of link key;
- link key exchange;
- authentication;
- generation of Encryption key in each device.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

The initialization key K_{init} depends on the identity of the devices. The key is generated with E_{22} algorithm, combining Bluetooth Device Address, PIN code, length of the PIN (in octets), and a random number. It is important to use this key only during initialization.
Link key K_{link} generation and exchange is performed as one procedure. The link key is generated as a combination key, dependent on two devices. Each device has its own random number; this number is used to generate half of the combination key with E_{21} algorithm. A random number is xor-ed with K_{init} and exchanged with another device. After receiving the number, the second half of the combination key is generated. Combination key K_{AB} is created as xor of both halves. If it consists of all zeroes, the key should be discarded. In another case, the new link key $K_{link} = K_{AB}$ is accepted.
Once the key is established, authentication shall be performed. Authentication uses E_1 and additionally outputs an Authenticated Ciphering Offset (ACO).
After this step, encryption key K_{enc} may be generated. The encryption key is the most interesting in terms of attacks presented in this paper. This key is derived by algorithm E_3 as follow:

$$K_{enc} = E_3(K_{link}, EN_{RAND}, COF), \tag{1}$$

where:

- K_{link} is a 128-bit current link key, common for both devices.
- EN_{RAND} is 128-bit publicly known random number;
- COF is 96-bit Ciphering Offset number. The number is determined in two ways. The offset is derived from the Bluetooth Device Address if the current link key is temporary. Otherwise, an authentic ciphering offset is used.

Algorithm E_3 is called each time encryption is activated. Moreover, K_{enc} shall be periodically refreshed. When E_0 cipher is used, refresh shall be done at least once every 2^{28} ticks of the Bluetooth clock (about 23.5 hours).

2.1 Encryption procedure concept

The encryption systems consist of three parts: session key generator, keystream generator, and part responsible for encryption and decryption. The first part takes specified inputs, mixes them, and shifts the output stream into the second part. The second part takes the output from the previous part as a seed and generates a keystream. The cipher should be re-initialized for each new packet with a maximum size of 2,745 bits. With each re-initialization, a new session key is generated.
Each device shall have the maximal allowed key length (in octets) $1 \leq L_{MAX} \leq 16$. Before encryption starts, the length of key L should be negotiated. Then, the encryption key is modified as follows:

$$K'_{enc} = g_2^L(x)(K_{enc}(x) \bmod g_1^L(x)), \tag{2}$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

where $g_1^L(x)$, $g_2^L(x)$ are polynomials defined for specified length L and $\deg(g_1^L(x)) = 8L$, $\deg(g_2^L(x)) \leq 128 - 8L$.
Inputs for the session key generator are as follows:

- $8L$-bit encryption key: K'_{enc};
- 48-bit Central's Bluetooth Device Address: ADR;
- 26-bit Central real-time clock: CL.

Such inputs provide that at least one bit is changed between two packets. Thus, each packet is encrypted with the new keystream.

2.2 Stream generation algorithm E_0 description

Algorithm E_0 is illustrated in Figure 1. Cipher has three major building blocks: linear feedback shift registers, summation combiner logic, and blend register. The generator consists of four different LFSRs with five taps, specified by the following polynomials $f_i(t)$:

- LFSR_1: $f_1(t) = t^{25} + t^{20} + t^{12} + t^8 + 1$;
- LFSR_2: $f_2(t) = t^{31} + t^{24} + t^{16} + t^{12} + 1$;
- LFSR_3: $f_3(t) = t^{33} + t^{28} + t^{24} + t^4 + 1$;
- LFSR_4: $f_4(t) = t^{39} + t^{36} + t^{28} + t^4 + 1$.

One bit from specified position of each LFSR, 24, 24, 32, 32, labeled as x_1, x_2, x_3, x_4 respectively, is connected with second part of generator. Summation Combiner Logic is built with classical xor gate, which produces output bits and function F_1, producing value $s_{t+1} = 2s_{t+1}^1 + s_{t+1}^0$, where $s_{t+1}^0, s_{t+1}^1 \in \mathbb{Z}_2$, for last part of generator. Each output bit is generated with the following equation:

$$z_t = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus c_t^0. \tag{3}$$

F_1 function is defined as follows:

$$F_1 : s_{t+1} = \left\lfloor \frac{\sum_{i=1}^{4} x_i + c_t}{2} \right\rfloor. \tag{4}$$

Last part of cipher is blend register holding two 2-bit values: $c_t = 2c_t^1 + c_t^0$ and $c_{t-1} = 2c_{t-1}^1 + c_{t-1}^0$, where $c_t^0, c_t^1, c_{t-1}^0, c_{t-1}^1 \in \mathbb{Z}_2$. The next value stored in the blend register is computed with a function F_2 using two different linear bijections, presented with equation (5).

$$\begin{aligned} &T_1 : (x_1, x_2) \to (x_1, x_0), \\ &T_2 : (x_1, x_2) \to (x_0, x_1 \oplus x_0). \end{aligned} \tag{5}$$

F_2 function is defined as follows:

$$F_2 : c_{t+1} = s_{t+1} \oplus T_1[c_t] \oplus T_2[c_{t-1}]. \tag{6}$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

Each value stored in registers can be threatened as a vector, then equation (6) can be presented as:

$$\begin{aligned} c_{t+1}^1 &= s_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^0, \\ c_{t+1}^0 &= s_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^1 \oplus c_{t-1}^0. \end{aligned} \tag{7}$$

Some simplifications can be applied in (4). This equation can be presented in an alternative form:

$$2s_{t+1}^1 + s_{t+1}^0 = \left\lfloor \frac{\sum_{i=1}^4 x_i + 2c_t^1 + c_t^0}{2} \right\rfloor.$$

Moreover, floor can be replaced with additional binary variable $\beta \in \mathbb{Z}_2$ in following way:

$$2s_{t+1}^1 + s_{t+1}^0 + \beta \cdot \frac{1}{2} = \frac{\sum_{i=1}^4 x_i + 2c_t^1 + c_t^0}{2}.$$

Finally, equation (4) can be presented as follow:

$$4s_{t+1}^1 + 2s_{t+1}^0 + \beta = \sum_{i=1}^4 x_i + 2c_t^1 + c_t^0. \tag{8}$$

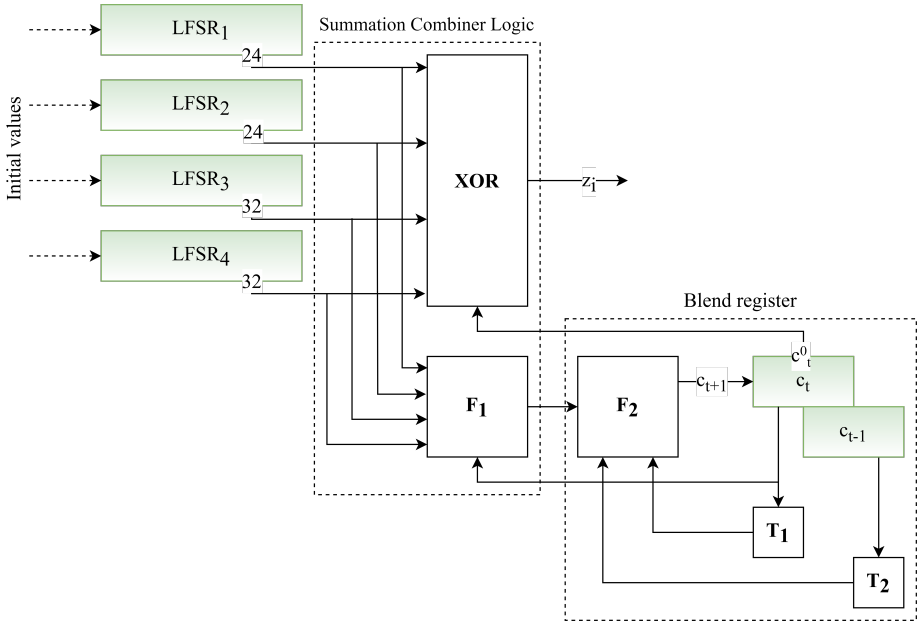

Fig. 1. Bluetooth encryption E_0 scheme. Based on [3].

Before the keystream is generated, initialization must be performed. Inputs are arranged as presented in Table 1. In following table, X_i denoted i-th bit of binary sequence X. As X[i] i-th octet of binary sequence is denoted.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

Table 1. Arranging the inputs for session key generator. Based on [3].

	Input						
$LFSR_1$	ADR[2]	CL[1]	$K'_{enc}[12]$	$K'_{enc}[8]$	$K'_{enc}[4]$	$K'_{enc}[0]$	CL_{24}
$LFSR_2$	ADR[3]	ADR[0]	$K'_{enc}[13]$	$K'_{enc}[9]$	$K'_{enc}[5]$	$K'_{enc}[1]$	$CL_3CL_2CL_1CL_0001$
$LFSR_3$	ADR[4]	CL[2]	$K'_{enc}[14]$	$K'_{enc}[10]$	$K'_{enc}[6]$	$K'_{enc}[2]$	CL_{25}
$LFSR_4$	ADR[5]	ADR[1]	$K'_{enc}[15]$	$K'_{enc}[11]$	$K'_{enc}[7]$	$K'_{enc}[3]$	$CL_7CL_6CL_5CL_4111$

Initialization lasts 240 steps. Each input, presented in Table 1, is shifted into the register starting from the rightmost (last column) bit. The state is updated without feedback as long as the first input bit does not reach the rightmost position of the specified LFRS. When the first bit reaches the rightmost position of the last LFRS, blend registers are reset $c_t = c_{t-1} = 0$. From this point, the output sequence is generated. The remaining input bits are shifted; when the last bit is shifted in, then 0 is put as input. After generating 240 output bits, the last 128 bits are loaded without updating the blend register to feedback registers.

3 Attack idea

The first version of Bluetooth supporting Secure Connections was presented in 2013, described in [2]. Until version 4.1 E_0, the cipher was used to ensure confidentiality. From version 4.1, the basic algorithm is AES-CCM. However, AES-CCM is used only if both devices support Bluetooth at least 4.1 version. Despite the years, devices using Bluetooth 4.0 and older are still popular. Devices using this standard are available for purchase in many places. Such a situation raises many possibilities for breach of confidentiality.

3.1 Current cryptanalysis

Significant papers on cryptanalysis of the E_0 cipher exist. Research divides attacks into two categories: attacking E_0 itself and attacking E_0 with assumptions in accordance with the specifications. When single level E_0 is attacked, only 2,745 output bits are produced with a single key. Previous results are presented in Table 2. In latest research [9], algebraic attack, which can be made with 2^{84} complexity, is presented. La Scala et al. show an attack requiring only 60 bits of keystream. The attack is realized by applying an algebraic attack twice on single E_0. Attack with the best complexity was presented in 2005 by Yi Lu et al. [10]. Conditional Correlation Attack requires $2^{23.8}$ frames and recovers encryption key with 2^{38} complexity. Despite the large number of packets needed, the attack is still practical, as mentioned in Section 2 encryption key is refreshed after 2^{28} clocks.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

Table 2. Previous attacks on Bluetooth encryption system.

	Time	Required data	Two level E_0	Additional requirements
Fluhrer, Lucks, 2000, [6]	2^{84}	140 bits	✓	-
	2^{77}	2^{30} bits	✓	
	2^{73}	2^{43} bits	✓	
Golić, Bagini, Morgari, 2002, [7]	2^{70}	45 packets	✓	Pre-computation sorting out a database of 2^{80} 103-bit words
Krause, 2002, [8]	$2^{76.8}$	128 bits	-	-
	$2^{112.95}$	128 bits	✓	-
Courtois, 2003, [5]	2^{49}	$2^{23.4}$ bits	-	2^{28} pre-computation
Armknecht, Krause, 2003, [1]	$2^{67.58}$	$2^{23.07}$ bits	-	-
Lu, Meier, Vaudenay, 2005, [10]	2^{38}	first 24 bits from $2^{23.8}$ packets	✓	-
Shaked, Wool, 2006, [12]	2^{87}	128 bits	✓	-
La Scala, Polese, Tiwari, Visconti, 2022, [9]	2^{84}	60 bits	✓	-

3.2 Proposed attack scheme

In this section, the proposed attack is presented. The main idea is to use quantum annealing to solve a system of equations describing the cipher like in [4]. It is important to note that quantum annealing can be used to solve optimization problems in specified form. In this paper, Quadratic Unconstrained Binary Optimization, formulated as Equation (9), is used,

$$\min_{x\in\{0,1\}^N} x^T Q x, \tag{9}$$

where x is binary variable vector, Q is upper-triangular $N \times N$ matrix containing real values. To obtain a cipher-based QUBO problem, algebraic representation of cipher with a system of equations (10) must be converted

$$\begin{cases} f_0(x_0, \ldots, x_{n-1}) \equiv 0 \pmod 2, \\ f_1(x_0, \ldots, x_{n-1}) \equiv 0 \pmod 2, \\ \vdots \\ f_{m-1}(x_0, \ldots, x_{n-1}) \equiv 0 \pmod 2. \end{cases} \tag{10}$$

The above system of equations can be transformed in the following way:

1. Transformation to equations f_i' with binary variables and integer coefficients:

$$f_i' \equiv 0 \pmod 2 \rightarrow f_i - 2k_i = 0,$$

where k_i is integer, $k \leq \left\lfloor \frac{f_i^{max}}{2} \right\rfloor$ and f_i^{max} is the maximal value of the polynomial f_i. It is important to note that seven equations are binary in

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

(11), and only the last equation has binary variables and integer values. The above remark simplifies transformation to QUBO problem for E_0 cipher. Only seven remaining equations must be converted.

2. Linearization $f'_i \rightarrow f'_{lin}$, with additional penalty Pen calculation. E_0 design induces no nonlinear equations, so in the case of this paper, $f'_i = f'_{lin}$ and $Pen = 0$.
3. Variables k_i replacement with binary variables $x_0, x_1, \ldots, x_{bl(k_i^{max})-1}$, where $bl(x)$ denotes bit-length of x and

$$k_i = \sum_{j=0}^{bl(k_i^{max})-2} 2^j x_j + (k_i^{max} - 2^{bl(k_i^{max})-1} + 1) \cdot x_{bl(k_i^{max})-1}.$$

4. Determination of polynomial $F'_{\text{Pen}} = \sum_{i=0}^{m-1} (f'_{\text{lin}_i})^2 + c \cdot Pen$, where c is a constant to prevent achieving the minimum energy for incorrect solutions.
5. Final form of the polynomial $F_{\text{Pen}} = F'_{\text{Pen}} - C$ calculation, where C is a constant term in F'_{Pen}. This constant corresponds to the minimum energy of the function.

The complexity of quantum annealing mainly depends on a number of variables used. The fewer variables used, the smaller the complexity and the easier to embed the problem.
Bluetooth encryption system design, presented in Section 2, requires the use of quantum annealing twice. Figure 2 presents the scheme of attack juxtaposed with the cipher design. The first stage of attack involves initial state recovery. Based on the keystream, a system of equations is generated. To generate equations, 129 bits of output are needed. For each bit, eight equations, presented as (11), are required.

$$\begin{cases} f_0 : z_t = l_{t+1} \oplus m_{t+7} \oplus n_{t+1} \oplus o_{t+7} \oplus c_t^0, \\ f_1 : l_{t+25} = l_t \oplus l_{t+5} \oplus l_{t+13} \oplus l_{t+17}, \\ f_2 : m_{t+31} = m_t \oplus m_{t+7} \oplus m_{t+15} \oplus m_{t+19}, \\ f_3 : n_{t+33} = n_t \oplus n_{t+5} \oplus n_{t+9} \oplus n_{t+29}, \\ f_4 : o_{t+39} = o_t \oplus o_{t+3} \oplus o_{t+11} \oplus o_{t+35}, \\ f_5 : c_{t+1}^1 = s_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^0, \\ f_6 : c_{t+1}^0 = s_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^1 \oplus c_{t-1}^0, \\ f_7 : 4s_{t+1}^1 + 2s_{t+1}^0 + \beta = l_{t+1} + m_{t+7} + n_{t+1} + o_{t+7} + 2c_t^1 + c_t^0. \end{cases} \tag{11}$$

The entire system of equations can be simplified, similarly to in [13]. The first simplification concerns the equations associated with LFSRs. Last keystream bit is associated with $l_{129}, m_{135}, n_{129}, o_{135}$. This remark allows limiting the number of equations for each keystream bit. Equations for remaining bits are not included in the system. In the final system of equations appears:

– 129 keystream-based equations f_0;
– 105 LFSR-based equations f_1 and f_2;

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

– 97 LFSR-based equations f_3 and f_4;
– 128 equations f_5, f_6 and f_7.

The second stage is divided into two parts. In the first part, the internal state must be recovered. Based on the initial state from the first stage, a system of equations is made. Successive bits are treated as a 128-bit output according to the order shown in the [3]. With such output, a system of equations, likewise the first stage, is made. Due to the smaller number of keystream bits, the whole system of equations looks slightly different than in the first stage. In the final system of equations appears:

– 128 keystream-based equations f_0;
– 104 LFSR-based equations f_1 and f_2;
– 96 LFSR-based equations f_3 and f_4;
– 127 equations f_5, f_6 and f_7.

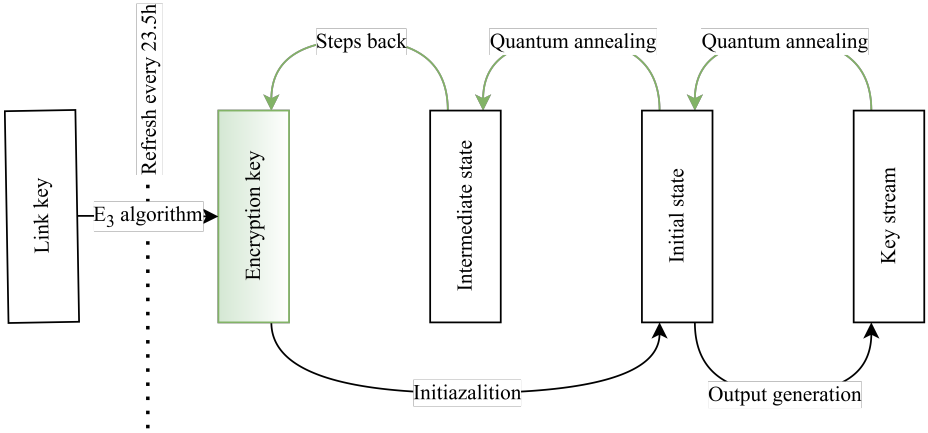

Fig. 2. Attack scenario of the proposed attack.

3.3 Probability of recovering used encryption key

However, recovering the Encryption key is possible using only a limited number of consecutive bits; even obtaining the ground state of the solved QUBO problem does not necessarily give the proper internal state and, therefore, the proper Encryption key. In this subsection, we focus on estimating how large is the probability of obtaining the proper encryption key using the attack scenario presented in the previous section.

To be able to make any estimations, at the beginning, we have to make the following assumptions:

1. The first assumption is that analyzed stream cipher behaves as the random generator.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

2. Even if the analyzed stream cipher behaves as a random generator, it is possible that the same keystream may be obtained for two or more different internal states. It means that there may be two or more proper ground states. It is worth noting that only one, the correct internal state, allows us to obtain the correct Encryption key. However, to make any estimations, it has to be assumed that the probability of obtaining every ground state is the same. In practical implementations, it is not obligatory because the shape of the energy landscape may favor some ground states against other ground states.

So, using the assumptions above, we may estimate the probability of obtaining the correct Encryption key.

Here, we use estimations from [13], where the probability of obtaining a proper internal state was estimated according to our assumptions. The most important thing here is the difference between the size of the internal state n and the number of known keystream bits k. More details may be found in [13]. The table showing the probability of obtaining a proper internal state is presented below.

Table 3. The probability of obtaining proper internal state according to the difference between the number of inner state bits n and the number of used consecutive keystream bits k.

$n-k$	3	2	1	0	-1	-2	-3	-4	-5	-6
Probability of obtaining proper ground state	0.12	0.25	0.43	0.63	0.79	0.88	0.94	0.97	0.98	>0.99

Now, because to recover the encryption key, we have to perform our attack twice, and in the second step, we cannot use more than 128 consecutive bits of the keystream, one cannot over gain the probability equal to 0.63 of obtaining the proper encryption key, but in this case in the first step the number of used consecutive bits of keystream should be equal at least to 136. In such a case, the attack complexity increases because, as defined in the first step, the QUBO problem is bigger than the QUBO problem in the second step.

Even though cryptanalysis often assumes that one should focus on attack parameters that give the success probability equal to at least 0.5. In such a case, the situation here is simple: in the first step, one has to use 129 consecutive keystream bits, while in the second step, one has to use 128 consecutive keystream bits. In such a case, the probability of obtaining the proper Encryption key equals $0,50$, which we wanted to obtain.

4 Results

Based on the proposed attack described above and the equations system (11), appropriate optimization problems were created in the form of QUBO for 129 and 128-bit keystreams. The obtained problems consist of 2,751 and 2,728 binary variables for 129 and 128 keystream bits, respectively.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

4.1 Graph of the obtained optimization problem

The Q matrix (see Equation (9)) of the obtained optimization problem, for a 129-bit keystream, has a degree of 2,751 and consists of 20,778 non-zero elements, whose values range from -16 to 17. For a 128-bit keystream, the Q matrix of the obtained QUBO problem has a degree of 2,728 and consists of 20,598 non-zero elements, the values of which are also in the range of -16 to 17. For both obtained problems, the number of non-zero elements is only 0.55% of all elements of matrix Q, which indicates that these matrices are sparse. The structure of the Q matrix of the obtained QUBO problem for the 129-bit keystream is presented in Figure 3, where the axes denote the indexes of the binary variables occurring in the objective function and the black point indicates the non-zero coefficient of a given quadratic monomial.

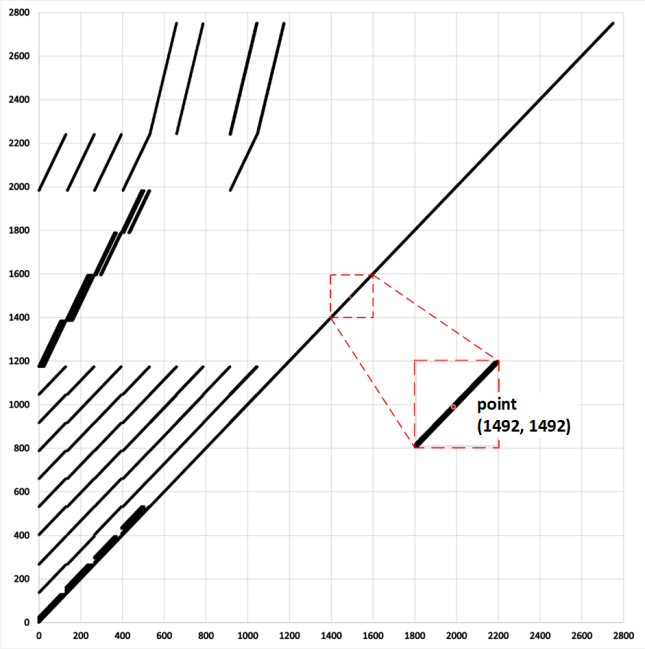

Fig. 3. The structure of the Q matrix of the obtained QUBO problem for the 129-bit keystream.

Since the QUBO problem is defined using an objective function with binary variables, it can be represented as a graph called the problem graph. A vertex represents each binary variable, and each quadratic monomial is represented by an edge between the given vertices.

For the obtained QUBO problems, the number of vertices in the problem graph is equal to the number of binary variables in an optimization problem,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

while the number of edges is 18,156 for a 129-bit keystream and 17,998 for a 128-bit keystream. For both QUBO problems, the maximum vertex degree is 40, and the remaining vertex degrees and the number of vertices of a given degree are shown in Table 4.

Table 4. Vertex degrees of the resulting graphs of QUBO problems.

Vertex degree	5	6	8	11	12	13	16	17	20	21	22	24
Number of vertices for 128-bit keystream	252	1,323	128	1	135	128	32	1	1	28	173	2
Number of vertices for 129-bit keystream	254	1,335	129	1	136	129	32	1	1	28	174	2

Vertex degree	26	27	28	29	30	32	33	34	36	38	40	
Number of vertices for 128-bit keystream	12	32	36	124	10	19	12	7	142	73	57	
Number of vertices for 129-bit keystream	12	32	36	125	10	19	12	7	144	74	58	

4.2 Embedding the problem graph in the hardware graph of the D-Wave quantum annealer

The structure of the D-Wave system's quantum processing unit (QPU) can be represented as a network of qubits connected using couplers. The network of qubits and the connecting couplers can be represented using the so-called hardware graph. Unfortunately, the hardware graph of the annealers currently provided by D-Wave is not complete, which has the greatest impact on the possibility of using them to solve any optimization problem. The most important parameters determining the possibility of solving a given problem are the number of physical qubits, the number of connections between qubits (couplers), and the arrangement of qubit connections (topology). D-Wave has developed different topologies for subsequent generations of quantum annealers. Currently, annealers are available with three topologies: Chimera, Pegasus, and Zephyr.

In the latest generation Zephyr topology, qubits are oriented horizontally and vertically and are connected by internal, external, and odd couplers. Inner couplers connect a pair of qubits oriented orthogonally to each other, external couplers connect a pair of qubits that are collinear, i.e., parallel in the same row or column, and odd couplers connect a pair of qubits parallel to each other in adjacent rows or columns. Two parameters characterize a given topology:

- nominal length of qubit, meaning the number of orthogonal qubits to which a given qubit is connected using internal couplers,
- degree of qubit, meaning the number of different qubits to which a given qubit is connected using all types of couplers.

The Zephyr topology consists of 71,736 couplers and 7,440 qubits, with a nominal length of 16 and a degree of 20.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

To solve an optimization problem using a D-Wave computer, the problem must be embedded in the QPU, i.e., the problem graph must be mapped in the hardware graph, which is called minor-embedding. During this process, each vertex from the problem graph is mapped to a physical qubit of the hardware graph and each edge to a coupler. As mentioned earlier, the hardware graphs of current topologies are not complete; therefore, to map all the connections of a given vertex, it may be necessary to map it not to one but to several physical qubits connected by couplers, creating a chain. All qubits of a given chain must return the same value in solution. Since the embedding problem is an NP-hard problem, its implementation in the currently available D-Wave system is performed using a heuristic tool, which results in the fact that for the same problem, different embedding results can be obtained using a different number of resources. An embedding can be characterized by the number of physical qubits used, the number of all couplers used, and the number and length of chains created.

Since the QUBO problem for an algebraic attack on the E0 cipher with a 128-bit keystream is too large to embed even in the latest Zephyr architecture, smaller problems were generated for the unchanged E0 algorithm but for a shorter keystream. The biggest problem embedded in the Zephyr architecture was generated for a 58-bit keystream. The graph of the embedded problem consists of 1,118 vertices and 6,938 edges. The maximum vertex degree is 38. This problem was mapped to a hardware graph consisting of 6,367 physical qubits and 12,187 couplers and required the creation of 811 chains ranging from 2 to 44 qubits in length. Table 5 shows the embedded problem's parameters and its embedding parameters.

Table 5. Parameters of the problem graph and hardware graph for 58-bit keystream.

Vertex degree	5	6	8	11	12	13	16	17	20	21	22	24	26	27
Number of vertices	112	483	58	1	65	58	40	1	1	31	108	2	12	29
Vertex degree	28	29	30	32	33	36	38							
Number of vertices	19	54	5	15	12	9	3							
Length of chain	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of chains	293	117	38	32	37	29	23	23	21	14	19	23	15	19
Length of chain	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Number of chains	13	7	15	12	7	2	5	1	6	6	3	1	3	2
Length of chain	30	31	32	33	34	36	37	38	39	40	41	42	43	44
Number of chains	1	2	3	1	2	3	1	1	2	1	3	2	1	2

4.3 Dedicated architecture of quantum annealer

It is known that D-Wave develops commercial quantum computers that solve optimization problems presented in the BQM (Binary Quadratic Model), CQM (Constrained Quadratic Model), or DQM (Discrete Quadratic Model) models.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

To enable solving as many different problems as possible, the developed quantum annealer topologies must maintain a kind of compromise between the number of physical qubits, the number of couplers, and their arrangement in the QPU unit, which shows the continuous development of the developed architectures.

In [13] the idea of constructing a quantum annealer only for algebraic attacks on a specific cipher was presented. In such a dedicated annealer, the emphasis would be on mapping the problem graph to the hardware graph as closely as possible, with the ideal situation being that the hardware graph would be the problem graph.

As previously presented, in the case of an algebraic attack on the E0 cipher, we need to solve two QUBO problems, one for a 128-bit keystream and the other for a 129-bit keystream. The QUBO problem for a 129-bit keystream is generated based on 916 equations of the form of Equation (11), of which 908 equations form an equations system based on which the QUBO problem for a 128-bit keystream is generated. Since the objective function of the QUBO problem is derived from the sum of squares, the QUBO problem for a 128-bit keystream is included in the QUBO problem for a 129-bit keystream. It follows that adding further equations generates new quadratic monomials without changing the existing ones, which means that new edges appear in the problem graph without changing the existing ones, but at most increasing the degree of the vertices. This means that the problem graph for a 128-bit keystream is a sub-graph of the problem graph for a 129-bit keystream. Therefore, it is enough to construct one dedicated annealer for the larger problem, to solve both.

A dedicated annealer, one-to-one mapping the problem graph to a hardware graph, would require 2,751 qubits with a maximum degree of 40 and 18,156 couplers. This value of the degree of qubit is higher than the value of the connectivity parameter in current quantum annealers. However, assuming a qubit degree of 20 (the same as in the latest D-Wave annealer topology) and any length and orientation of the qubits, for an algebraic attack on the E0 cipher, the hardware graph would have to consist of 3,542 qubits, 18,947 couplers, and 733 chains, including 675 chains length 2 and 58 chains length 3. It is worth emphasizing here that the amounts of required resources are smaller than the resources currently offered by D-Wave annealers.

5 Conclusion

In this paper was presented the algebraic attack on E0 cipher using quantum annealing. In compare to other attacks on cryptograpihic algorithms using quantum annealing, presented attack requires the smallest number of logical variables from all ciphers with 128-bits of keylength.

It is worth to note, that in some cases, presented attack scenario allows one to retrieve encryption key used for 23,5 hours at most, therefore it is possible to decipher the (almost) whole daily communication performing algebraic attack using quantum annealing twice. The first attack uses 129 consecutive keystream bits - this attack allows to obtain 128 bits of inner state, which is also treated as

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28

the keystream used in the second attack. If in both attacks one obtains proper ground state as the solution, in such a case there is 50% probability that obtained encryption key will be proper.

However the complexity of algebraic attacks using quantum annealing still is not well known, assuming attack complexity as $O(e^{\sqrt{N}})$ [11], where N is the number of logical variables, the complexity of presented attack for obtaining encryption key is much below the brute force attack complexity and may be estimated as $2^{76.52}$.

References

1. Armknecht, F., Krause, M.: Algebraic attacks on combiners with memory. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 162–175. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
2. Bluetooth Special Interest Group: Bluetooth Core Specification (2013), rev. 4.1
3. Bluetooth Special Interest Group: Bluetooth Core Specification (2021), rev. 5.3
4. Burek, E., Wroński, M., Mańk, K., Misztal, M.: Algebraic attacks on block ciphers using quantum annealing. IEEE Transactions on Emerging Topics in Computing **10**(2), 678–689 (2022)
5. Courtois, N.T.: Fast algebraic attacks on stream ciphers with linear feedback. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 176–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
6. Fluhrer, S.R., Lucks, S.: Analysis of the e0 encryption system. In: ACM Symposium on Applied Computing (2001), https://api.semanticscholar.org/CorpusID:2130499
7. Golić, J.D., Bagini, V., Morgari, G.: Linear cryptanalysis of bluetooth stream cipher. In: Knudsen, L.R. (ed.) Advances in Cryptology — EUROCRYPT 2002. pp. 238–255. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
8. Krause, M.: Bdd-based cryptanalysis of keystream generators. In: Knudsen, L.R. (ed.) Advances in Cryptology — EUROCRYPT 2002. pp. 222–237. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
9. La Scala, R., Polese, S., Tiwari, S.K., Visconti, A.: An algebraic attack to the bluetooth stream cipher e0. Finite Fields and Their Applications **84**, 102102 (2022). https://doi.org/https://doi.org/10.1016/j.ffa.2022.102102, https://www.sciencedirect.com/science/article/pii/S1071579722001113
10. Lu, Y., Meier, W., Vaudenay, S.: The conditional correlation attack: A practical attack on bluetooth encryption. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. pp. 97–117. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
11. Mukherjee, S., Chakrabarti, B.K.: Multivariable optimization: Quantum annealing and computation. The European Physical Journal Special Topics **224**(1), 17–24 (2015)
12. Shaked, Y., Wool, A.: Cryptanalysis of the bluetooth e0 cipher using obdd's. Cryptology ePrint Archive, Paper 2006/072 (2006), https://eprint.iacr.org/2006/072, https://eprint.iacr.org/2006/072
13. Wroński, M., Burek, E., Leśniak, M.: (in)security of stream ciphers against quantum annealing attacks on the example of the grain 128 and grain 128a ciphers. Cryptology ePrint Archive, Paper 2023/1502 (2023), https://eprint.iacr.org/2023/1502, https://eprint.iacr.org/2023/1502

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_28