Transformation of the discrete logarithm problem over $\mathbb{F}_{2^n}$ to the QUBO problem using normal bases

Michał Wroński, and Mateusz Leśniak

Abstract—**Quantum computations are very important branch of modern cryptology. According to the number of working physical qubits available in general-purpose quantum computers and in quantum annealers, there is no coincidence, that nowadays quantum annealers allow to solve larger problems. In this paper we focus on solving discrete logarithm problem (DLP) over binary fields using quantum annealing. It is worth to note, that however solving DLP over prime fields using quantum annealing has been considered before, no author, until now, has considered DLP over binary fields using quantum annealing. Therefore, in this paper, we aim to bridge this gap. We present a polynomial transformation of the discrete logarithm problem over binary fields to the Quadratic Unconstrained Binary Optimization (QUBO) problem, using approximately $3n^2$ logical variables for the binary field $\mathbb{F}_{2^n}$. In our estimations, we assume the existence of an optimal normal base in the given fields. Such a QUBO instance can then be solved using quantum annealing.**

Keywords—**Discrete Logarithm Problem, Quantum Annealing, Binary Fields**

I. INTRODUCTION

SHOR's algorithm [1] has long been considered the most promising method for solving classical cryptography problems such as integer factorization, the discrete logarithm problem in finite fields, and the elliptic curve discrete logarithm problem.

In 2023, Shor's algorithm was improved by Regev [2], and later by Ragavan and Vaikuntanathan [3] in the case of integer factorization, and then by Ekerå and Gärtner in the case of the discrete logarithm problem in finite fields [4].

However, it is worth noting that other quantum methods have become more prominent with the existing quantum hardware. Taking this into account, quantum annealing is now considered the most practical method of quantum computation, which allows for computing some instances of the discrete logarithm problem (DLP) over finite fields that are larger than those solvable using Shor's algorithm [5], [6].

It is worth noting that the application of quantum annealing to cryptanalysis has been considered previously. The first paper, presented by Jiang in [7], demonstrated the use of quantum annealing for the cryptanalysis of the factorization problem. Subsequently, other cryptanalysis problems have

been explored, such as breaking the discrete logarithm [5], [6], [8]–[10] and attacking symmetric ciphers [11]–[16].

As shown above, while computing the discrete logarithm problem over prime fields using quantum annealing has been considered before, no author has addressed DLP over binary fields using quantum annealing.

In this paper, we aim to bridge this gap. We present a polynomial transformation of the discrete logarithm problem over binary fields to the Quadratic Unconstrained Binary Optimization (QUBO) problem, using approximately $3n^2$ logical variables for the binary field $\mathbb{F}_{2^n}$. In our estimations, we assume the existence of an optimal normal base in the given fields. Such a QUBO instance can then be solved using quantum annealing.

Interestingly, this transformation requires more logical variables than a similar transformation for DLP over a prime field $\mathbb{F}_p$, whose bit-length is equal to n. The main reason is the complexity of transforming binary field arithmetic into a pseudo-Boolean function. Therefore, there are many additional variables, making such a transformation less efficient than in the case of prime fields.

II. DLP IN $\mathbb{F}_{2^n}$ USING OPTIMAL NORMAL BASES

This section focuses on the fields $\mathbb{F}_{2^n}$. We assume the use of such $\mathbb{F}_{2^n}$ fields where an optimal normal basis exists.

For the field $\mathbb{F}_{2^n}$, the normal basis consists of elements $\{\alpha^{2^0}, \alpha^{2^1}, \ldots, \alpha^{2^{n-1}}\}$, whereas in the commonly used polynomial basis, elements form the set $\{1, \beta, \ldots, \beta^{n-1}\}$. We assume here that the field $\mathbb{F}_{2^n}$ is generated by the irreducible polynomial $f(t)$ of degree n.

It is worth noting that if in the given field exists the optimal normal basis of the II type, then it is always possible to find such a polynomial $f(t)$ of degree n that $\alpha = t$ is the generator of the normal basis in the field $\mathbb{F}_{2^n}$.

Such an irreducible polynomial can be constructed recursively using Dickson polynomials in the following manner [17]:

$$\begin{cases} f_0(t) = 1, \\ f_1(t) = t + 1, \\ f_n(t) = t \cdot f_{n-1}(t) - f_{n-2}(t), \quad n \geq 2. \end{cases} \quad (1)$$

A normal basis is optimal if its multiplication matrix T consists of $2n - 1$ nonzero elements.

M. Wroński is with Department of Cryptology, NASK - National Research Institute, Warsaw, Poland (e-mail: michal.wronski@nask.pl).

M. Leśniak is with Department of Cryptology, NASK - National Research Institute, Warsaw, Poland (e-mail: mateusz.lesniak@nask.pl).

For the simplicity of our estimations, in the subsequent sections, we assume that an optimal normal basis exists for the given binary field $\mathbb{F}_{2^n}$. Therefore, in the multiplication matrix T, in $n-1$ rows (columns), there are just two elements "1", and in one row (column), there occurs just one "1". This assumption will be important in the analysis of the complexity of our problem.

Let us make the following assumption: the field $\mathbb{F}_{2^n}$ is generated by the irreducible polynomial $f(t)$, given by the Dickson polynomial, for such a field an optimal normal basis of II type exists, and the generator of the multiplicative subgroup of this field is t.

In such a case, one can define the multiplication matrix $T^{(0)}$, which allows one to obtain the least significant bit c_0 of the resulting register C. However, by making rotations of columns and rows, it is also possible to obtain other bits of register C (see [18]).

Therefore, the multiplication of any two elements $A, B \in \mathbb{F}_{2^n}$ (presented as vectors) is performed in the following manner:

$$C = A \cdot T^{(0)} \cdot B^T. \tag{2}$$

Then the product C, generally, may be given by the following system of equations:

$$\begin{aligned} c_k = \quad & a_{(i_0+k \mod n)} b_{(j_0+k \mod n)} \\ + \quad & (a_{(i_1+k \mod n)} b_{(j_1+k \mod n)} \\ + \quad & a_{(j_1+k \mod n)} b_{(i_1+k \mod n)}) \\ + \quad & \dots \\ + \quad & (a_{(i_{n-2}+k \mod n)} b_{(j_{n-2}+k \mod n)} \\ + \quad & a_{(j_{n-1}+k \mod n)} b_{(i_{n-1}+k \mod n)}). \end{aligned} \tag{3}$$

Now, let us assume that the generator of the multiplicative subgroup is t.

We begin the main part of this section by defining the discrete logarithm problem, similarly as in [5], [6]:

$$t^y = h, \tag{4}$$

in the multiplicative subgroup of the field $\mathbb{F}_{2^n}$, so $t, h \in \mathbb{F}_{2^n}^*$ and $y \in \{1, \dots, 2^n - 1\}$. This problem is equivalent to:

$$t^y \equiv h \mod f(t), \tag{5}$$

for elements $t, h \in \mathbb{F}_{2^n}^*$ and $y \in \{1, \dots, 2^n - 1\}$.

Let us note that the bit length of $2^n - 1$ equals n. Noting that y can be written using n bits, and if $y = 2^{n-1}u_n + \dots + 2u_2 + u_1$, where $u_1, \dots, u_n$ are binary variables, then

$$t^y = t^{2^{n-1}u_n} \cdots t^{2u_2} t^{u_1}. \tag{6}$$

Let us also note that

$$A \cdot t^{2^{i-1}u_i} = \begin{cases} A, & \text{if } u_i = 0, \\ A \cdot t^{2^{i-1}}, & \text{if } u_i = 1. \end{cases} \tag{7}$$

So now let us assume that we have to perform the multiplication of $C = A \cdot t^{2^{l-1}u_l}$.

Using Equation (3), one obtains the following result:

$$c_k = a_i u_l + a_j u_l + a_k(1 - u_l). \tag{8}$$

Let us note that the equation above is correct. When $u_l = 1$, then $C = A \cdot t^{2^{l-1}}$, and therefore in the normal basis representation, only bit b_l is equal to 1, and others are equal to 0. In the opposite situation, when $u_l = 0$, then $C = A \cdot 1$. However, in normal bases representation, $1 = t + t^2 + \dots + t^{2^{n-1}}$. Note that in Equation (8), both situations are considered. If $u_l = 0$, then $c_k = a_k$ for every $k = \overline{0, n-1}$, which results in $C = A$, and therefore it is equivalent to multiplication by 1. Otherwise, if $u_l = 1$, then $c_k = a_i u_l + a_j u_l$, where at least one of a_i, a_j is non-zero (we assume that we always perform multiplication by of non-zero elements). Therefore, C is the result of multiplication of register A by $t^{2^{l-1}}$.

III. EXAMPLE ANALYSIS FOR DLP OVER $\mathbb{F}_{2^5}$ AND ITS GENERALIZATION

To better illustrate the transformation of the discrete logarithm problem over binary fields to the QUBO problem, we will demonstrate this process using the small degree extension field $\mathbb{F}_{2^5}$. Using such a small field should clarify the general method for transforming DLP over binary fields to the QUBO problem.

Consider the given field $\mathbb{F}_{2^5}$. For such a field, there exists an optimal normal basis of type II [18].

The irreducible polynomial $f(t)$, which generates the field $\mathbb{F}_{2^5}$, is obtained using formula (1) and is given as $f(t) = t^5 + t^4 + t^2 + t + 1$.

When using the normal basis, the multiplication matrix $T^{(0)}$ is given [18] as:

$$T^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{9}$$

Now let $A, B, C \in \mathbb{F}_{2^5}$, where $C = A \cdot B$. Then, the product C is given by

$$\begin{aligned} c_0 = \quad & a_4 b_4 + (a_0 b_1 + a_1 b_0) + (a_1 b_3 + a_3 b_1) \\ + \quad & (a_2 b_4 + a_4 b_2) + (a_2 b_3 + a_3 b_2), \end{aligned}$$

$$\begin{aligned} c_1 = \quad & a_0 b_0 + (a_1 b_2 + a_2 b_1) + (a_2 b_4 + a_4 b_2) \\ + \quad & (a_3 b_0 + a_0 b_3) + (a_3 b_4 + a_4 b_3), \end{aligned}$$

$$\begin{aligned} c_2 = \quad & a_1 b_1 + (a_2 b_3 + a_3 b_2) + (a_3 b_0 + a_0 b_3) \\ + \quad & (a_4 b_1 + a_1 b_4) + (a_4 b_0 + a_0 b_4), \end{aligned} \tag{10}$$

$$\begin{aligned} c_3 = \quad & a_2 b_2 + (a_3 b_4 + a_4 b_3) + (a_4 b_1 + a_1 b_4) \\ + \quad & (a_0 b_2 + a_2 b_0) + (a_0 b_1 + a_1 b_0), \end{aligned}$$

$$\begin{aligned} c_4 = \quad & a_3 b_3 + (a_4 b_0 + a_0 b_4) + (a_0 b_2 + a_2 b_0) \\ + \quad & (a_1 b_3 + a_3 b_1) + (a_1 b_2 + a_2 b_1). \end{aligned}$$

Now, let us assume that the generator of the multiplicative subgroup is t. We begin the main part of this section by defining the discrete logarithm problem, similarly as presented in Equations (4) and (5).

For any $A \in \mathbb{F}_{2^5}$ given as $a_4 t^{2^4} + a_3 t^{2^3} + a_2 t^{2^2} + a_1 t^{2^1} + a_0 t^{2^0}$, let us set, for example, $B = t^{2^0 u_0}$. In such a case, the result C of the multiplication of $A \cdot B$, will be given as:

$$
\begin{aligned}
c_0 =& \quad a_1 u_0 + a_0(1 + u_0), \\
c_1 =& \quad a_0 u_0 + a_3 u_0 + a_1(1 + u_0), \\
c_2 =& \quad a_3 u_0 + a_4 u_0 + a_2(1 + u_0), \quad (11) \\
c_3 =& \quad a_2 u_0 + a_1 u_0 + a_3(1 + u_0), \\
c_4 =& \quad a_4 u_0 + a_2 u_0 + a_4(1 + u_0).
\end{aligned}
$$

Detailed analysis shows that for every field $\mathbb{F}_{2^n}$, the resulting register C will have a similar form for multiplying A by any $B = t^{2^i u_i}$, for $\overline{0, n-1}$. More precisely, $n-1$ bits of register C will consist of 4 monomials: 2 monomials of degree 2 occur because in the multiplication matrix $T^{(0)}$ there are two "1"s, one monomial of degree 2, and one monomial of degree 1, because there is multiplication of a_i by $(1+u_j)$, for $\overline{0, n-1}$. One bit of register C will consist of 3 monomials: 1 monomial of degree 2 occurs because in the multiplication matrix $T^{(0)}$ there is one "1", and because there is multiplication of a_i by $(1 + u_j)$, for $\overline{0, n-1}$, one monomial of degree 2, and one monomial of degree 1.

Therefore, we may estimate the total number of variables required for transforming DLP over $\mathbb{F}_{2^n}$ to the QUBO problem. We use the same decomposition tree as in [5].

For a single node, there will be necessary n binary variables for register C. Moreover, there are also necessary n new variables for linearization (note that monomials of degree two of the form $u_i a_j$, where $j = \overline{0, n-1}$ will occur). There will also be necessary $2(n-1)$ new variables for k (this is necessary for $n - 1$ bits), and for one bit of register C, one additional variable for k will be necessary. Of course, one variable is necessary to represent u_i. Summing up, there will be necessary $4n$ variables for a single node. As we have approximately n nodes, the total number of variables will equal approximately $4n^2$.

However, the amount of variables described above ($4n^2$) may be lowered. Let us note that each of the single equations c_i can be transformed into a pseudo-boolean function in the following manner (let us take, for example, c_1):

$$c_1 + a_0 u_0 + a_3 u_0 + a_1(1 + u_0) = 0.$$

As the maximal value of the left side of the equation is equal to 5, two new variables are necessary for k. However, note that in binary notation, the equation above is equivalent to

$$-c_1 + a_0 u_0 + a_3 u_0 + a_1(1 - u_0) = 0.$$

Now one can transform this equation into the pseudo-boolean function as

$$-c_1 + a_0 u_0 + a_3 u_0 + a_1(1 - u_0) - 2k_1 = 0,$$

where $k_1 \in \{0, 1\}$. Why is it possible? Let's observe that the maximal value of $-c_1 + a_0 u_0 + a_3 u_0 + a_1(1 - u_0)$ is equal to 2, and the minimal value is equal to -1. It means that $0 \leq k_1 \leq 1$. Therefore, we require only one bit for multiplicity representation instead of two bits, as was described above.

It is important to see that a similar trick may be done for all fields $\mathbb{F}_{2^n}$, for which optimal normal bases exist. In such a case, transforming the DLP problem over $\mathbb{F}_{2^n}$ will require $3n^2$ logical variables. However, whether this transformation may be obtained using fewer variables is unknown.

IV. WORKING EXAMPLE

In this section the practical example of application of our method will be presented.

A. Normal bases and multiplication matrix definition over $\mathbb{F}_{2^3}$

According to the formula (1), the irreducible polynomial over $\mathbb{F}_{2^3}$, for which the generator of the optimal normal basis of II type is $\alpha = t$ is $f(t) = t^3 + t^2 + 1$. However, using basic algebraic properties of normal and polynomial bases, we can check that it is true.

So now we will check if for the field $\mathbb{F}_{2^3}$ generated by the polynomial $f(t) = t^3 + t^2 + 1$, the generator of the normal basis will be the element $\alpha = t$. To do this, note that:

Normal basis elements	Polynomial basis elements
α	t
α^2	t^2
α^4	$t^4 = t^2 + t + 1$

To check if α is indeed the normal basis generator, we will try to create the transition matrix between the normal basis and the polynomial basis.

Assume that b_2, b_1, b_0 are the coefficients of an element in the normal basis (the element is of the form $b_2\alpha^4 + b_1\alpha^2 + b_0\alpha$), while a_2, a_1, a_0 are the coefficients in the polynomial basis (the element is of the form $a_2 t^2 + a_1 t + a_0$). Then the transition from the normal basis to the polynomial basis can be done as follows:

$$
\begin{bmatrix} a_2 & a_1 & a_0 \end{bmatrix}
\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}
= \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix}
\begin{bmatrix} \alpha^4 \\ \alpha^2 \\ \alpha \end{bmatrix} =
$$

$$
\begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix}
\begin{bmatrix} t^2 + t + 1 \\ t^2 \\ t \end{bmatrix}
= \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix}
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}.
$$

From above, it results that:

$$
\begin{bmatrix} a_2 & a_1 & a_0 \end{bmatrix}
\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}
= \begin{bmatrix} b_2 + b_1 & b_2 + b_0 & b_2 \end{bmatrix}
\begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}.
$$

Thus, it can be immediately noted that $\alpha = t$ will be the generator of the normal basis because the transition matrix from the normal basis to the polynomial basis

$$
M_{N \to P} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
$$

is non-singular.

The transition matrix from the polynomial basis to the normal basis is:

$$
M_{P \to N} = M_{N \to P}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}.
$$

The result of multiplying two elements

$$A = a_2\alpha^4 + a_1\alpha^2 + a_0\alpha$$

and

$$B = b_2\alpha^4 + b_1\alpha^2 + b_0\alpha$$

is:

$$a_2b_2\alpha^8 + (a_1b_2 + a_2b_1)\alpha^6 + (a_0b_2 + a_2b_0)\alpha^5 + a_1b_1\alpha^4 + (a_0b_1 + a_1b_0)\alpha^3 + a_0b_0\alpha^2.$$

Note that:

$$\begin{aligned}
\alpha^8 &= t, \\
\alpha^6 &= t^2 + t, \\
\alpha^5 &= t + 1, \\
\alpha^4 &= t^2 + t + 1, \\
\alpha^3 &= t^2 + 1, \\
\alpha^2 &= t^2.
\end{aligned} \tag{12}$$

Thus, the result of the multiplication can be written using the polynomial basis as

$$(a_1b_2 + (a_2 + a_1 + a_0)b_1 + (a_1 + a_0)b_0) t^2 +$$

$$+ ((a_2 + a_1 + a_0)b_2 + (a_2 + a_1)b_1 + a_2b_0) t +$$

$$+ a_0b_2 + (a_1 + a_0)b_1 + (a_2 + a_1)b_0 =$$

$$c_2't^2 + c_1't + c_0'.$$

Using the transition matrix from the polynomial basis to the normal basis, we obtain that

$$c_2't^2 + c_1't + c_0' = c_0'\alpha^4 + (c_2' + c_0')\alpha^2 + c_1' + c_0' = c_2\alpha^4 + c_1\alpha^2 + c_0.$$

Thus, $c_2 = c_0'$, $c_1 = c_2' + c_0'$, and $c_0 = c_1' + c_0'$. We then get that

$$c_0 = a_2b_2 + a_1b_2 + a_2b_1 + a_0b_1 + a_1b_0,$$

$$c_1 = a_0b_0 + a_2b_0 + a_0b_2 + a_1b_2 + a_2b_1,$$

$$c_2 = a_1b_1 + a_0b_1 + a_1b_0 + a_2b_0 + a_0b_2.$$

Thus, we have obtained the multiplication matrices.

$$T^{(0)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix},$$

$$T^{(1)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

$$T^{(2)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

So now, let us consider the multiplicative subgroup of field $\mathbb{F}_{2^3}$ generated by irreducible polynomial $f(t) = t^3 + t^2 + 1$. Let $g = t$ be the generator of this subgroup. Let $h = t^4 + t^2$. We will show how to transform this problem to the QUBO form.

B. Transformation of the example problem

First, let us look that the order of multiplicative subgroup $\mathbb{F}_{2^3}^*$ is equal to 7, which is prime. Therefore, we have to solve the following problem:

$$g^y \equiv h \mod (f(t)), \tag{13}$$

which is equivalent to the problem of solving

$$t^y \equiv t^4 + t^2 \mod (t^3 + t^2 + 1), \tag{14}$$

where $y = 4u_2 + 2u_1 + u_0$, for binary variables u_0, u_1, u_2. We use normal bases instead of commonly used polynomial bases.

Let us note that it is equivalent to

$$t^{4u_2 + 2u_1 + u_0} = t^{4u_2}t^{2u_1}t^{u_0} = $$
$$t^{4u_2}t^{2u_1}t^{u_0} \equiv t^4 + t^2 \mod (t^3 + t^2 + 1). \tag{15}$$

As we use normal basis system representation, one can use vector notation in which $t = [0, 0, 1]$ and t^2 and t^4 are simply rotations of t. In such a case $t^2 = [0, 1, 0]$ and $t^4 = [1, 0, 0]$. Let us note that the neutral element in normal basis is $t^4 + t^2 + t = [1, 1, 1]$.

Let us take the multiplication matrix $T^{(0)}$, defined as:

$$T^{(0)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{16}$$

Now let $A, B, C \in \mathbb{F}_{2^3}$, where $C = A \cdot B$. Then, product C is given by

$$\begin{aligned}
c_0 &= a_2b_2 + a_1b_2 + a_2b_1 + a_0b_1 + a_1b_0, \\
c_1 &= a_0b_0 + a_2b_0 + a_0b_2 + a_1b_2 + a_2b_1, \\
c_2 &= a_1b_1 + a_0b_1 + a_1b_0 + a_2b_0 + a_0b_2.
\end{aligned} \tag{17}$$

Now, let us note that

$$t^{u_0} = \begin{cases} 1 = t + t^2 + t^4, u_0 = 0, \\ t, u_0 = 1. \end{cases} \tag{18}$$

So t^{u_0} may be presented as:

$$t^{u_0} = t + (1 - u_0)t^2 + (1 - u_0)t^4 = [1 - u_0, 1 - u_0, 1]. \tag{19}$$

Writing t^{u_0} in general form and using new variables, one obtains $t^{u_0} = v_{0,0}t + v_{0,1}t^2 + v_{0,2}t^4$, where $v_{0,0} = 1, v_{0,1} = 1 - u_0, v_{0,2} = 1 - u_0$.

Now let's perform the multiplication of t^{u_0} by t^{2u_1}. Similarly as before, if $u_1 = 0$, then the result will be equal to $t^{u_0} = [v_{0,2}, v_{0,1}, v_{0,0}]$. If $u_1 = 1$, one must use Equation (17). In such a case the resulting vector $[v_{1,2}, v_{1,1}, v_{1,0}]$ will be of the following form:

$$\begin{cases} v_{1,0} = v_{0,2}u_1 + v_{0,0}u_1 + (1 - u_1)v_{0,0}, \\ v_{1,1} = v_{0,2}u_1 + (1 - u_1)v_{0,1}, \\ v_{1,2} = v_{0,1}u_1 + v_{0,0}u_1 + (1 - u_1)v_{0,2}. \end{cases} \tag{20}$$

Now, let's note that the system of equations above must be considered while analyzing the transformation of the DLP over $\mathbb{F}_{2^3}$ to the QUBO problem.

Finally, the last step is multiplication of $[v_{1,2}, v_{1,1}, v_{1,0}]$ by t^{4u_2}. This step goes as follows:

$$\begin{cases} v_{2,0} = v_{1,2}u_2 + v_{1,1}u_2 + (1-u_2)v_{1,0} = 0, \\ v_{2,1} = v_{1,0}u_2 + v_{1,1}u_2 + (1-u_2)v_{1,1} = 1, \\ v_{2,2} = v_{1,0}u_2 + (1-u_2)v_{1,2} = 1. \end{cases} \quad (21)$$

But let's note that because $h = t^4 + t^2$, then the vector $[v_{2,2}, v_{2,1}, v_{2,0}]$ is equal to $[1, 1, 0]$.

Now, we transform and simplify the system of equations given above. We set $u_3 = v_{0,0}, u_4 = v_{0,1}, u_5 = v_{0,2}, u_6 = v_{1,0}, u_7 = v_{1,1}, u_8 = v_{1,2}, v_{2,0} = 0, v_{2,1} = 1, v_{2,2} = 1$.

Let us note that first:

$$\begin{cases} u_3 = 1, \\ u_4 = 1 - u_0, \\ u_5 = 1 - u_0 = u_4. \end{cases} \quad (22)$$

Then

$$\begin{cases} u_6 = u_5 u_1 + u_3 u_1 + (1-u_1)1 = \\ u_4 u_1 + u_3 u_1 + 1 - u_1 = u_4 u_1 + 1, \\ u_7 = u_4 u_1 + (1-u_1)u_4 = u_4, \\ u_8 = u_4 u_1 + u_1 + (1-u_1)u_4 = u_1 + u_4. \end{cases} \quad (23)$$

And then

$$\begin{cases} 0 = u_2 u_8 + u_2 u_7 + (1-u_2)u_6 = u_2 u_8 + u_2 u_4 + (1-u_2)u_6, \\ 1 = u_2 u_6 + u_2 u_7 + (1-u_2)u_7 = u_2 u_6 + u_7 = u_2 u_6 + u_4, \\ 1 = u_2 u_6 + (1-u_2)u_8 = u_2 u_6 + u_8 - u_2 u_8. \end{cases} \quad (24)$$

Taking into account the whole equations, making at first linearization, removing all terms on the left side and making squaring, and, finally, adding at the end penalty, one can obtain the final QUBO form of our problem of solving DLP over binary fields. This method is presented in detail, for example, in [5] and [6]. The necessary equations to prepare the final QUBO problems are given below. Let us note that in the system below, using arithmetic tricks, the number of additional variables necessary to represent the multiplicities of 2 after transforming the equation from the boolean function to the pseudo boolean function is lowered:

$$\begin{cases} F_1 = (1 - u_0 - u_4)^2, \\ F_2 = (1 - u_6 - u_{10})^2, \\ F_3 = (u_8 + u_1 + u_4 - 2u_{14})^2, \\ F_4 = (-u_{11} + u_{12} + u_6 - u_{13})^2, \\ F_5 = (1 - u_{13} - u_4)^2, \\ F_6 = (1 - u_{13} - u_8 + u_{11})^2, \\ Pen_1 = Penalty(u_1, u_4, u_{10}), \\ Pen_2 = Penalty(u_2, u_8, u_{11}), \\ Pen_3 = Penalty(u_2, u_4, u_{12}), \\ Pen_4 = Penalty(u_2, u_6, u_{13}), \end{cases} \quad (25)$$

where Pen is a standard penalty in Rosenberg form: when is made substitution $z = xy$, the resulting penalty is of the form $(xy - 2(x+y)z + 3z)$ and is obtained by invoking function $Penalty(x, y, z)$. Then, the final QUBO problem is given by

$$F = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + Pen_1 + Pen_2 + Pen_3 + Pen_4. \quad (26)$$

We transformed the problem above into a QUBO problem using only 11 logical variables. Using quantum annealing, we obtained the correct solution of $y = 5$.

The scheme presenting embedding of our problem into the D-Wave Advantage 2 prototype 2.3 system having 1248 qubits is presented in Figure 1.
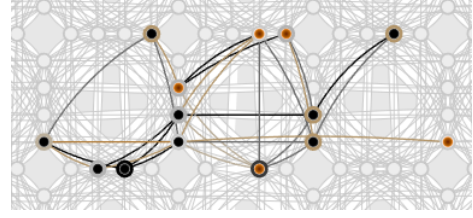

Fig. 1. Embedding of DLP for $\mathbb{F}_{2^3}$ in the D-Wave Advantage 2 QPU.

The problem and used solver parameters are presented in Table I.

TABLE I
D-WAVE ADVANTAGE SOLVER PARAMETERS USED IN SOLVING QUBO PROBLEM EQUIVALENT TO THE PROBLEM OF FINDING DISCRETE LOGARITHM OVER $\mathbb{F}_{2^3}$ IN THE SUBGROUP OF SIZE 7.

Parameter	Value
Name (chip ID)	Advantage2_prototype2.3
Available qubits	1,248
Topology	Zephyr
Number of reads	10,000
Annealing time (μs)	20
Anneal schedule	[[0,0],[20,1]]
H gain schedule	[[0,0],[20,1]]
Programming thermalization (μs)	1000

TABLE II
RESULTS PARAMETERS OBTAINED DURING SOLVING QUBO PROBLEM EQUIVALENT TO THE PROBLEM OF FINDING DISCRETE LOGARITHM OVER $\mathbb{F}_{2^3}$ IN THE SUBGROUP OF SIZE 7.

Parameter	Value
Number of target variables	14
Max chain length	2
Chain strength	1.8827
QPU access time (μs)	804,627.61
QPU programming time (μs)	19,227.61
QPU sampling time (μs)	785,400
Total post-processing time (μs)	1
Post processing overhead time (μs)	1

We ran the problem above using quantum annealing 10,000 times. 7,415 trials gave proper minimal energy, which means that for the given example, the probability of obtaining the proper result is equal to 74.15%.

V. PROBABILITY ANALYSIS

In this section we will analyze the probability of obtaining correct solution using quantum annealing. We will show that, contrary to Shor's algorithm application, in our case it is much easier to show that our results are not random.

It is worth to note that experiments of solving small DLPs instances over prime fields have been conducted using Shor's

algorithm. As far as we know, the biggest succesfully solved experiment [19] was solving:

$$2^x \equiv 1(mod\ 3). \tag{27}$$

However, even for such small example it was not trivial to show that application of Shor's algorithm worked correctly and wasn't the result of random computations.

We begin by calculating how many minimal solutions our QUBO problem (26) has.

As in many cases there are possible many equivalent proper solutions, in our case there exists only one proper solution. Let us note that because the proper solution is $y = 5$, it means that it is the only proper solution of exponent. It is worth to note that if proper solution would be $y = 0$, then also $y = 7$ would be proper solution of our problem, because $7 \mod 7 = 0$.

Now we have to check if there may be equivalent proper solutions, where only values of multiplicities $k's$ differ. It is worth to note that if for example the some value k_i may be at most equal to 2, then in such a case $k = u_i + u_j$ for some indices i, j and binary variables u_i and u_j. But let's note that if proper solution is $k = 1$, then such solution may be obtained on two different ways. The first solution will be for $u_i = 1, u_j = 0$, while the second will be for $u_i = 0, u_j = 1$.

However, let us look that in the definition of our QUBO problem there is only one multiplicity k (in our case it is only u_{14}), and it is written using only 1 bit. Therefore, only one minimal energy solution is in our case possible.

Now we will show that running our experiment using quatnum annealing, obtaining the minimal energy solution randomly is practically impossilble. So we will estimate the probability of obtaining proper solution using quantum annealing. Let's note that, using for example Shor's algorithm, the space is much smaller and therefore, the probability distribution of different possible values (proper and also unproper) looks more uniformly. Therefore, in Shor's algorithm often a lot of effort goes to analysis if obtaining proper solution has appropriate big probability and if it may be distinguished from the random solution.

Below we prove that our solution is not obtained randomly. Let's note that our QUBO problem consists of 11 logical variables. It means that the solution space is $2^{11} = 2048$. In this case, only one solution is proper, which means that, using binomial distribution, the probability of single success is equal to $\frac{1}{2048}$.

We made $10,000$ trials. Assuming that from the cryptographic point of wiev we may assume that method works if returns the proper solution with probability 0.5, we will compute how much likely is that one obtains at least 5000 successes, making 10000 trials, if obtaining each state would be as same probable. In such a case one has to compute cumulated binomial distribution

$$\sum_{i=5000}^{10000} \binom{1000}{i} \left(\frac{1}{2048}\right)^i \left(\frac{2047}{2048}\right)^{10000-i}. \tag{28}$$

It is worth noting that such probability is extremely small and is equal to approximately $3.103 \cdot 10^{-13550}$.

As in our experiment we obtained 7,415 successes in 10,000 trials, it is clear that our experiment is not a random one, therefore it is statistically proved that our method works correctly and gives almost 75% probability of success in the presented example.

VI. CONCLUSION

In this paper the application of quantum annealing to solve discrete logarithm problem over $\mathbb{F}_{2^n}$ fields has been presented. It is worth to note that our method of transformation of DLP to the QUBO problem requires approximately $3n^2$ logical variables if for given field the optimal normal basis exists. We presented the experiment where DLP over small binary field ($\mathbb{F}_{2^3}$) was solved using quantum annealing. Our experiment has almost 75% probability of returning proper result for such small field.

However existing of optimal normal basis in the given field is restrictive assumption, one should note that for many binary fields such optimal normal bases exist. If for a given field such optimal normal basis does not exist, then the presented method of transformation of the DLP to the QUBO problem would require asymptotically more resources.

It is however unknown how far we can go on the current quantum annealers, because we did not conducted such experiments for larger fields. Further works should cover this gap and solving of larger instances be tried.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134. [Online]. Available: https://doi.org/10.1109/SFCS.1994.365700

[2] O. Regev, "An efficient quantum factoring algorithm," 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2308.06572

[3] S. Ragavan and V. Vaikuntanathan, "Space-efficient and noise-robust quantum factoring," in *Advances in Cryptology – CRYPTO 2024*, L. Reyzin and D. Stebila, Eds. Cham: Springer Nature Switzerland, 2024, pp. 107–140. [Online]. Available: https://doi.org/10.1007/978-3-031-68391-6_4

[4] M. Ekerå and J. Gärtner, "Extending Regev's factoring algorithm to compute discrete logarithms," in *Post-Quantum Cryptography*, M.-J. Saarinen and D. Smith-Tone, Eds. Cham: Springer Nature Switzerland, 2024, pp. 211–242. [Online]. Available: https://doi.org/10.1007/978-3-031-62746-0_10

[5] M. Wroński, "Practical solving of discrete logarithm problem over prime fields using quantum annealing," in *Computational Science – ICCS 2022*. Cham: Springer International Publishing, 2022, pp. 93–106. [Online]. Available: https://doi.org/10.1007/978-3-031-08760-8_8

[6] M. Wroński, E. Burek, Ł. Dzierzkowski, and O. Żołnierczyk, "Transformation of elliptic curve discrete logarithm problem to qubo using direct method in quantum annealing applications," *Journal of Telecommunications and Information Technology*, no. 1, pp. 75–82, 2024. [Online]. Available: https://doi.org/10.26636/jtit.2024.1.1463

[7] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, "Quantum annealing for prime factorization," *Scientific Reports*, vol. 8, no. 1, p. 17667, Dec 2018. [Online]. Available: https://doi.org/10.1038/s41598-018-36058-z

[8] M. Wroński, "Index calculus method for solving elliptic curve discrete logarithm problem using quantum annealing," in *Computational Science – ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2021, pp. 149–155. [Online]. Available: https://doi.org/10.1007/978-3-030-77980-1_12

[9] O. Żołnierczyk and M. Wroński, "Searching b-smooth numbers using quantum annealing: Applications to factorization and discrete logarithm problem," in *Computational Science – ICCS 2023*, J. Mikyška, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds. Cham: Springer Nature Switzerland, 2023, pp. 3–17. [Online]. Available: https://doi.org/10.1007/978-3-031-36030-5_1

[10] M. Wroński and Ł. Dzierzkowski, "Base of exponent representation matters-more efficient reduction of discrete logarithm problem and elliptic curve discrete logarithm problem to the qubo problem," *Quantum Information and Computation*, vol. 24, no. 7&8, pp. 0541–0564, 2024. [Online]. Available: https://doi.org/10.26421/QIC24.7-8-1

[11] E. Burek, M. Wroński, K. Mańk, and M. Misztal, "Algebraic attacks on block ciphers using quantum annealing," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 678–689, 2022. [Online]. Available: https://doi.org/10.1109/TETC.2022.3143152

[12] E. Burek and M. Wroński, "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, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2022, pp. 143–149. [Online]. Available: https://doi.org/10.1007/978-3-031-08760-8_12

[13] M. Wronski, E. Burek, and M. Lesniak, " (In)Security of Stream Ciphers Against Quantum Annealing Attacks on the Example of the Grain 128 and Grain 128a Ciphers ," *IEEE Transactions on Emerging Topics in Computing*, no. 01, pp. 1–14, Oct. 5555. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/TETC.2024.3474856

[14] E. Burek, K. Mańk, and M. Wroński, "Searching for an efficient system of equations defining the aes sbox for the qubo problem," *Journal of Telecommunications and Information Technology*, vol. 94, no. 4, p. 30–37, Oct. 2023. [Online]. Available: https://jtit.pl/jtit/article/view/1340

[15] M. Leśniak, E. Burek, and M. Wroński, "Unsafe mechanisms of bluetooth, e_0 stream cipher cryptanalysis with quantum annealing," in *Computational Science – ICCS 2024*, L. Franco, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer Nature Switzerland, 2024, pp. 389–404. [Online]. Available: https://doi.org/10.1007/978-3-031-63778-0_28

[16] G. Morse, T. Kozsik, O. Mencer, and P. Rakyta, "A compact qubo encoding of computational logic formulae demonstrated on cryptography constructions," 2024. [Online]. Available: https://arxiv.org/abs/2409.07501

[17] R. C. Mullin and A. Mahalanobis, *Dickson bases and finite fields*. Faculty of Mathematics, University of Waterloo, 2005.

[18] A. Amin and T. F. Al-Somani, "Hardware implementations of gf ($2^{\hat{}}$ m) arithmetic using normal basis," *Journal of Applied Sciences*, vol. 6, no. 6, pp. 1362–1372, 2006. [Online]. Available: https://doi.org/10.3923/jas.2006.1362.1372

[19] Y. AONO, S. LIU, T. TANAKA, S. UNO, R. V. METER, N. SHINOHARA, and R. NOJIMA, "The present and future of discrete logarithm problems on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–21, 2022. [Online]. Available: https://doi.org/10.1109/TQE.2022.3183385