Practical solving of discrete logarithm problem over prime fields using quantum annealing*

Michał Wroński[0000−0002−8679−9399]

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

Abstract. This paper investigates how to reduce discrete logarithm problem over prime fields to the QUBO problem to obtain as few logical qubits as possible. We show different methods of reduction of discrete logarithm problem over prime fields to the QUBO problem. In the best case, if n is the bitlength of a characteristic of the prime field $\mathbb{F}_p$, there are required approximately $2n^2$ logical qubits for such reduction. We present practical attacks on discrete logarithm problem over the 4-bit prime field $\mathbb{F}_{11}$, over 5-bit prime field $\mathbb{F}_{23}$ and over 6-bit prime field $\mathbb{F}_{59}$. We solved these problems using D-Wave Advantage QPU. It is worth noting that, according to our knowledge, until now, no one has made a practical attack on discrete logarithm over the prime field using quantum methods.

Keywords: discrete logarithm problem, D-Wave Advantage, quantum annealing

1 Introduction

Shor's quantum algorithm for factorization and discrete logarithm computation [9] is one of the essential researches in modern cryptology. Since then, there have been many efforts to build a general-purpose quantum computer that solves real-world cryptographic problems. Unfortunately, till now, such powerful general-purpose quantum computers do not exist. On the other hand, quantum annealing is an approach that takes more and more popularity. The most powerful computer using quantum annealing technology is the D-Wave Advantage computer. One of the most exciting applications of quantum annealing to cryptography is transforming the factorization algorithm into the QUBO problem and then solving this problem using the D-Wave computer [6].

Moreover, the newest D-Wave computers have much more physical qubits than general-purpose quantum computers. It is believed that this approach also may be helpful, primarily until large general-purpose quantum computers will exist. It seems that, in some cases, D-Wave computers may be used to solve

* 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."

cryptographic problems, which cannot be solved nowadays by general-purpose quantum computers.

This paper shows how to transform discrete logarithm problem (DLP) over prime fields to the QUBO problem. We consider different approaches to such transformation, aiming to obtain the smallest possible number of logical qubits. The best method allows one to convert discrete logarithm problem to the QUBO problem using approximately $2n^2$ logical qubits.

Our contribution is:

- presenting different methods of reduction of discrete logarithm problem to the QUBO problem, where the best method requires approximately $2n^2$ logical qubits for such reduction;
- presenting practical attacks on discrete logarithm problem over the 4-bit prime field $\mathbb{F}_{11}$, over 5-bit prime field $\mathbb{F}_{23}$ and over 6-bit prime field $\mathbb{F}_{59}$ using D-Wave Advantage QPU.

It is worth noting that, according to our knowledge, until now, no one has made a practical attack on discrete logarithm over the prime field using quantum methods.

2 Quantum annealing and cryptography

Shor's quantum algorithm for factorization and discrete logarithm began the race to construct a quantum computer to solve real-world cryptographic problems. Nowadays, the two approaches of quantum computing for cryptography are the most popular.

The first approach is quantum annealing, used in D-Wave computers. The second approach is general-purpose quantum computing. The important thing is that the first approach has limited applications, where mainly QUBO and Ising problems may be solved using such quantum computers.

QUBO (Quadratic Unconstrained Boolean Optimization) [2] is a significant problem with many real-world applications. One can express the QUBO model by the following optimization problem:

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

where Q is an $N \times N$ upper-diagonal matrix of real weights, x is a vector of binary variables. Moreover, diagonal terms $Q_{i,i}$ are linear coefficients, and the nonzero off-diagonal terms are quadratic coefficients $Q_{i,j}$.

QUBO problem may also be viewed as a problem of minimizing the function

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

What is essential from the cryptological point of view, many problems may be translated to the QUBO problem. The most exciting example of such transformation is integer factorization. It is worth noting that the quantum factorization record had belonged to the D-Wave computer for some time. Using transformation of integer factorization to the QUBO problem, Dridi and Alghassi [3] factorized integer $200,099$, which result was later beaten by Jiang et al. [6], and by Wang et al. [11], who factorized 20-bit integer $1,028,171$. It is worth noting that quantum annealing was also used to find relations in the index calculus method for elliptic curves where using D-Wave Leap and hybrid sampler, elliptic curve discrete logarithm problem over the 8-bit prime field has been solved [12].

On the other hand, general-purpose quantum computers have limited resources. The most powerful Intel, Google, and IBM quantum computers have 49, 72, and 127 qubits, respectively [5], [4], [10]. It means that the resources of general quantum computers are nowadays too small to solve real-world cryptographic problems.

The D-Wave computers using quantum annealing are developing rapidly and have many more qubits than a few years before. The most potent quantum annealing computer, D-Wave Advantage [1], has $5,760$ working qubits. This quantum annealer allows solving general problems with up to $1,000,000$ variables and dense problems with $20,000$ variables. A detailed description of how the D-Wave computer works may be found in [2].

2.1 Quantum annealing and factorization problem

As presented in the introduction, quantum annealing may solve the factorization problem. We will shortly describe how to transform the factorization problem into the QUBO problem. For simplicity, we use the description of the direct method presented in [6].

Let $N = pq$, where p and q are prime numbers. Let l_1, l_2 be the bitlengths of p and q respectively. Because p and q are prime numbers, they can be written as $p = (x_{l_1-1}, x_{l_1-2}, \ldots, x_1, 1)_2$ and $q = (x_{l_1+l_2-2}, x_{l_1+l_2-3}, \ldots, x_{l_1}, 1)_2$, where $l_1 \geq l_2$ and $x_i \in \{0,1\}$ for $i = 1$ to $l_1 + l_2 - 2$. Let us define cost function as $f\left(x_1, x_2, x_3, \ldots, x_{l_1+l_2-2}\right)^2 = (N - pq)^2$. Multiplication of the binary representations for p and q yields a sum of binary products.

Now we will show how the resulting 3-local terms may be reduced to 2-local terms. Let us note that each penalty monomial of the form $x_i x_j x_l$ will be transformed, according to [6], in the following way

$$x_i x_j x_l \to u_k x_l + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k). \tag{3}$$

It means that

$$x_i x_j x_l = u_k x_l = u_k x_l + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k), \tag{4}$$

if $x_i x_j = u_k$ and

$$x_i x_j x_l < x_l u_k + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k), \tag{5}$$

if $x_i x_j \neq u_k$.

It results in that the term $x_ix_jx_l$ may be transformed to quadratic form by replacing x_ix_j with u_k plus a constraint, given by penalty term:

$$min(x_ix_jx_l) = min\left(x_lu_k + 2(x_ix_j - 2u_k(x_i + x_j) + 3u_k)\right). \tag{6}$$

It is worth noting that for integer factorization it is required $\binom{l_1-1}{2} + \binom{l_2-1}{2}$ auxiliary variables to form a quadratic cost function. If $l = l_1 = l_2 \frac{n}{2}$, the number of auxiliary variables is equal to $(l-1)(l-2)$ and $L = l(l-1)$ variables in total, which is approximately equal to $\frac{n^2}{4}$ where n is the bitlength of N. What is more, one can set $p = (1, x_{l_1-2}, \ldots, x_1, 1)_2$ and $q = (1, x_{l_1+l_2-4}, \ldots, x_{l_1}, 1)_2$ when lengths of p and q are known.

After transforming the factorization problem into the QUBO problem, it can be solved using, for example, quantum annealing on a D-Wave computer.

3 Methods of transformation of discrete logarithm problem to the QUBO problem

This section will present different approaches to transforming the discrete logarithm problem to the QUBO problem.

We begin by defining discrete logarithm problem

$$g^y = h, \tag{7}$$

in the prime field $\mathbb{F}_p$, where $g, h \in \mathbb{F}_p^*$ and $y \in \{1, \ldots, Ord(g)\}$. This problem is equivalent to

$$g^y \equiv h(mod\ p), \tag{8}$$

for integers $g, h \in \{1, \ldots, p-1\}, y \in \{1, \ldots, Ord(g)\}$.

Let m be the bitlength of $Ord(g)$. We begin by making the following transformation. Let us note that y may be written using m bits and if $y = 2^{m-1}u_m + \cdots + 2u_2 + u_1$, where $u_1, \ldots, u_m$ are binary variables, then

$$g^y = g^{2^{m-1}u_m + \cdots + 2u_2 + u_1} = g^{2^{m-1}u_m} \ldots g^{2u_2}g^{u_1}, \tag{9}$$

It is worth noting that writing $y = 2^{m-1}u_m + \cdots + 2u_2 + u_1$ allows to obtain $y > Ord(g)$, but because we operate in a cyclic group, one can always get the result from $\{1, \ldots, Ord(g)\}$ computing $y \mod Ord(g)$.

Let us also note that

$$g^{2^{i-1}u_i} = \begin{cases} 1, u_i = 0, \\ g^{2^{i-1}}, u_i = 1, \end{cases} \tag{10}$$

which is equivalent to

$$g^{2^{i-1}u_i} = 1 + u_i\left(g^{2^{i-1}} - 1\right). \tag{11}$$

Now we use the observation above to define different transformation approaches of discrete logarithm problem over prime fields to the QUBO problem.

3.1 Transformation of discrete logarithm problem to the QUBO problem - brutal approach

Let us note that using Equations (9) and (11), one obtains the following equation

$$\begin{aligned} g^y &= \left(1+u_m\left(g^{2^{m-1}}-1\right)\right)\cdot\dots\cdot\left(1+u_2\left(g^2-1\right)\right)\left(1+u_1\left(g-1\right)\right) \\ &= \left(1+u_m\left((g^{2^{m-1}}-1)mod\ p\right)\right)\dots\left(1+u_2\left((g^2-1)mod\ p\right)\right) \\ &\cdot\left(1+u_1\left((g-1)mod\ p\right)\right). \end{aligned} \tag{12}$$

We can see that g^y can be represented as the polynomial of degree m of m Boolean variables. We will show how to linearize this polynomial. Let us note that linearization may be performed in the following way.

If $m=1$, then $1+u_1(g-1)$ and it is indeed linear polynomial.

If $m=2$, then $f=(1+u_1(g-1))\left(1+u_2(g^2-1)\right)=1+u_1(g-1)+u_2(g^2-1)+u_1u_2(g-1)\left(g^2-1\right)$. The variable u_1u_2 may be substituted by an auxiliary variable $v_1=u_1u_2$. The penalty will be added later. So one can see that $f=1+u_1(g-1)+u_2(g^2-1)+v_1(g-1)\left(g^2-1\right)$ and is in linear form.

We can keep on such a procedure, and finally, one obtains the linear polynomial of 2^m-1 variables.

Having polynomial f in linear form, now we should transform modular equation $f\equiv h(mod\ p)$ to the equation over integers

$$(f-h)\ mod\ p-kp=0, \tag{13}$$

where $k\in\mathbb{Z}$ and for every polynomial f, operation $f\ mod\ p$ is equivalent to the reduction of all of the coefficients of polynomial f modulo p.

If one wants to solve Equation (13) searching for minimal energy of optimization problem, it is necessary to square Equation (13), obtaining in result polynomial F and Equation (14).

$$F=\left((f-h)\ mod\ p-kp\right)^2=0. \tag{14}$$

Let us note that k is bounded by the maximal number of monomials appearing in the polynomial $(f-h)\ mod\ p$, which is equal to 2^m. Finally, $k_{max}\leq\lfloor\frac{2^m(p-1)}{p}\rfloor<2^m$ and bitlength of k is equal to m at most. Moreover, to obtain proper energy, we have to add penalties to the function F, according to Equation (3), obtaining $F_{Pen}=F+Pen$, where Pen are penalties obtained during linearization. Polynomial F_{Pen} has minimal energy equal to 0 (our QUBO problem is constructed so that minimal energy is equal to 0 because in our QUBO problem appears constant energy offset).

The DLP transformation to the QUBO problem using a brutal approach requires, in general, 2^m+m-1 variables for m-bit order of an element g. The number of variables in such a case does not depend on the bit length of p. Let us note that this exponential growth makes that the presented method may be applied only for small prime fields $\mathbb{F}_p$.

3.2 Transformation of discrete logarithm problem to the QUBO problem using approximately $2n^2$ logical qubits - efficient approach.

In this section, we will transform the discrete logarithm problem to the QUBO problem using a regular binary tree of maximal height for decomposition. It is possible to obtain an equivalent QUBO problem using approximately $2n^2$ logical qubits in such a case, where n is the bitlength of a characteristic of the prime field $\mathbb{F}_p$. Let us define for every $i = \overline{1, m}$ equality $x_i = g^{2^{i-1} u_i}$. The scheme of such a regular binary tree of maximal height for general m (the bitlength of $Ord(g)$) used for decomposition of discrete logarithm problem to the QUBO problem is presented in Figure 1.

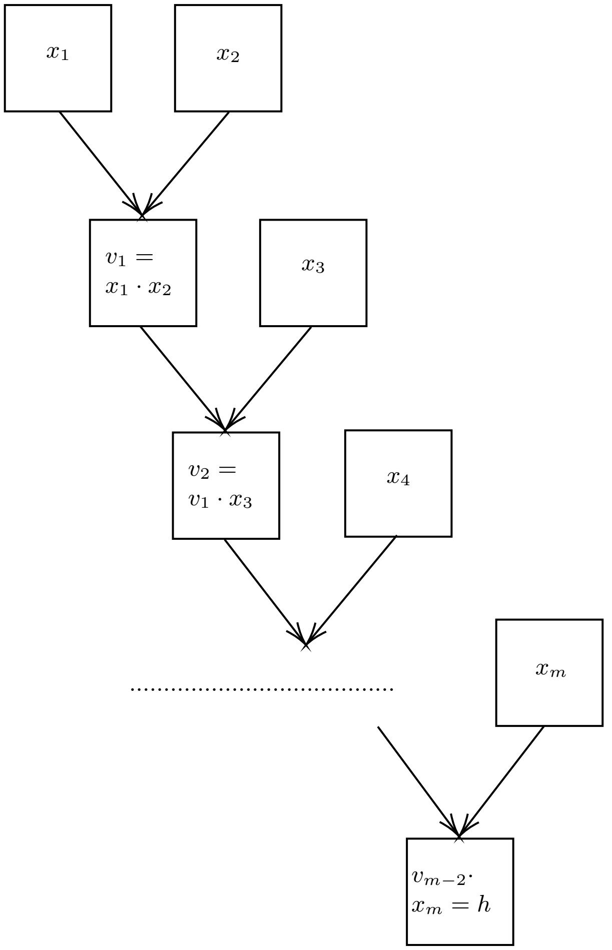


Fig. 1: The scheme of decomposition of discrete logarithm problem using the efficient approach.

It is worth noting that other decomposition methods are possible, as a method using a binary balanced tree. Unfortunately, in such a case the expected number of logical variables of equivalent QUBO problem is equal to approximately $\frac{n^3}{2}$. We, therefore, do not describe here this method.

At first, let us note that using the problem decomposition scheme presented in the Figure 1, in every step we can create a new variable $v_i = v_{i-1}x_{i+1}$, which is equivalent to $v_i \equiv v_{i-1}x_{i+1} (mod\ p)$. It is also easy to show that the total number of new variables v_i will be equal to $m-2$, because $x_1, \ldots, x_m$ are leaves of the binary tree with $m-1$ inner nodes, where each inner node is equivalent to some auxiliary variable. However, the root is not equivalent to any auxiliary variable, but it is equivalent to $v_{m-2}x_m \equiv h(mod\ p)$, so the number of auxiliary variables v_i is equal to $m-2$.

What is more, each of the equations for $v_1, v_2, v_{m-2}, \ldots, v_{m-3}, v_{m-2}, v_{m-2}x_m$ need to be transformed to the equation over integers.

$$\begin{cases} f_1 = (v_1 - x_1x_2)\ mod\ p - k_1p = 0, \\ f_2 = (v_2 - v_1x_3)\ mod\ p - k_2p = 0, \\ \ldots \\ f_{m-3} = (v_{m-3} - v_{m-4}x_{m-2})\ mod\ p - k_{m-3}p = 0, \\ f_{m-2} = (v_{m-2} - v_{m-3}x_{m-1})\ mod\ p - k_{m-2}p = 0, \\ f_{m-1} = (h - v_{m-2}x_m)\ mod\ p - k_{m-1}p = 0. \end{cases} \tag{15}$$

We will precisely count how many auxiliary variables are necessary. Let us note that for variables $v_1, \ldots, v_{m-2}$ it is necessary n bits to represent v_i, because $v_i \in \{1, \ldots, p-1\}$ and at most $n+1$ bits to represent k_i, because $(v_i - v_{i-1}x_{i+1})\ mod\ p$, according to Equation (11) is equivalent to

$$\left(v_i - v_{i-1} - v_{i-1}u_{i+1}\left(g^{2^i} + 1\right)\right) mod\ p. \tag{16}$$

Let us note that v_i is limited by its definition by $p-1$. Using the binary representation of $-v_{i-1}$ and making reduction modulo p, one obtains polynomial of binary variables and coefficients from interval $\{0, \ldots, p-1\}$. It means that maximal value of polynomial (16) is equal to $(2n+1)(p-1)$ and therefore, $kp \leq (2n+1)(p-1)$, which means that $k \leq \frac{(2n+1)(p-1)}{p} < 2n+1$, so $k \leq 2n$ and the bit length of k is equal to $\lfloor \log_2 (2n) \rfloor + 1$ at most.

Additionally, for every $i = \overline{1, m-2}$, during linearization of $(v_i - v_{i-1}x_{i+1})\ mod\ p$ it is necessary to linearize terms appearing in $v_{i-1}x_{i+1}$, which requires n variables, because, according to Equation (11), $x_{i+1} = 1 + u_{i+1}\left(g^{2^i} - 1\right)$ has two monomials but depends on only one variable.

Let us denote as $f_{lin_1}, \ldots, f_{lin_{n-1}}$ polynomials $f_1, \ldots, f_{m-1}$ after linearization. Then the final polynomial F in QUBO form is equal to

$$F_{Pen} = (f_{lin_1})^2 + \cdots + (f_{lin_{m-1}})^2 + Pen, \tag{17}$$

where Pen are penalties obtained during linearization and minimal energy of F_{Pen} is equal to 0.

So the total number of variables is equal:

- for $x_1, \ldots, x_m$ - it is required to have m binary variables,
- for $v_1, \ldots, v_{m-2}$ - it is required to have $(m-2)n$ binary variables,
- for $k_1, \ldots, k_{m-1}$ - it is required to have $(m-1)\left(\lfloor \log_2(2n) \rfloor + 1\right)$ binary variables,
- for auxiliary variables obtained during linearization of each polynomial $f_1, \ldots f_{m-1}$ it is required $(m-1)n$ variables,

Finally, obtained QUBO problem requires $m+(m-2)n+(m-1)\left(\lfloor \log_2(2n) \rfloor + 1\right) + (m-1)n = 2mn + 2m - 3n + (m-1)\lfloor \log_2(2n) \rfloor - 1$, which is approximately equal to $2mn$ variables. If we also assume that $m \approx n$ (what is true if the given generator is the generator of the multiplicative subgroup of field $\mathbb{F}_p$), then the total number of variables is equal to approximately $2n^2$.

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

1. one can at first linearize each equation f_i to obtain the linearized equation f_{Lin_i}, then compute the sum $F_{Pen} = \sum_{i=1}^{w} f_{Lin_i}^2 + Pen$, where Pen denotes penalties obtained during linearization; polynomial F_{Pen} is in such a case in QUBO form;
2. one can at first compute the sum $F = \sum_{i=1}^{w} f_i^2$, and then make of quadratization of the polynomial F, obtaining F_{Quadr}, finally obtaining polynomial $F_{Pen} = F_{Quadr} + Pen$ in QUBO form, where Pen denotes penalties obtained during quadratization.

In our methods of reducing discrete logarithm problem to the QUBO problem, the first method simply allows us to compute the maximal number of required variables in the resulting QUBO problem. The second method allows one often to obtain a smaller number of variables than the first method. Even though, in practical experiments, often the second method was used.

3.3 Mixed approach

In the mixed approach, the crucial observation is that, especially for small prime fields, the brutal approach is more efficient than approaches using binary tree decomposition. The key idea is to use both methods to obtain the problem using fewer logical qubits.

In the first step we multiply k first terms $x_1, x_2, \ldots, x_k$, as same as in the brutal approach, obtaining equation $(v_1 - x_1 \cdot x_2 \cdots x_k) \mod p - k_1 p = 0$. We use the decomposition method of discrete logarithm problem using a regular binary tree of maximal height in all following steps. The scheme of such decomposition is presented in Figure 2.

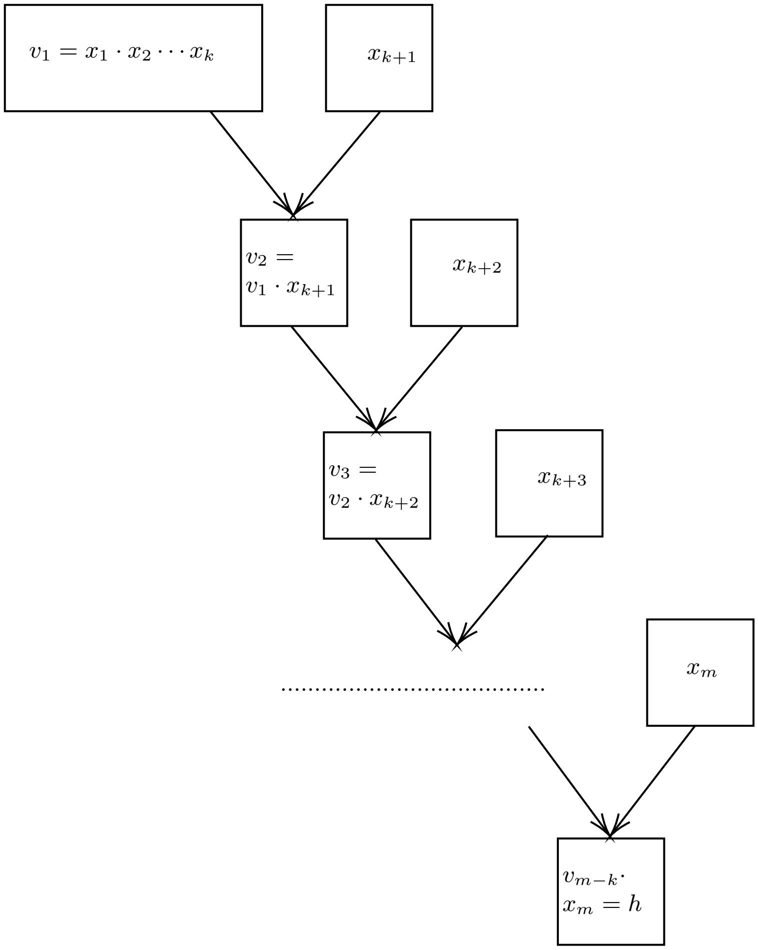

$v_1 = x_1 \cdot x_2 \cdots x_k$
x_{k+1}
$v_2 =$ $v_1 \cdot x_{k+1}$
x_{k+2}
$v_3 =$ $v_2 \cdot x_{k+2}$
x_{k+3}
x_m
$v_{m-k} \cdot$ $x_m = h$

Fig. 2: The scheme of decomposition of discrete logarithm problem using the mixed approach.

If choosing the number of variables we like to multiply in the first step is made carefully, the final QUBO problem may consist of a smaller number of variables than any other presented approach.

4 Experiments

We experimented with different approaches and solvers: classical, hybrid, and quantum. However, we aimed to solve discrete logarithm problem over prime fields using quantum solver for D-Wave Advantage QPU. In every case, as characteristic of n-bit prime field $\mathbb{F}_p$ we chose the biggest n-bit prime, for which $d = \frac{p-1}{2}$ is prime. We also chose as generators elements of order equal to d. The most notable result of solving discrete logarithm problems over the prime field using D-Wave Advantage QPU was solving discrete logarithm problem over the 4-bit prime field $\mathbb{F}_{11}$ using the efficient approach, 5-bit prime field $\mathbb{F}_{23}$ using the mixed approach and over 6-bit prime field $\mathbb{F}_{59}$ using the brutal approach.

4.1 Solving discrete logarithm problem over $\mathbb{F}_{11}$ using efficient approach

In the case of the prime field $\mathbb{F}_{11}$, the smallest QUBO problem can be obtained using the brutal approach, but we wanted to show the application of the efficient approach for solving discrete logarithm problem.

In this experiment, we solved the discrete logarithm problem over a field $\mathbb{F}_{11}$ which is the 4-bit prime field. The given generator was 4, and the order of 4 in $\mathbb{F}_{11}$ is 5. We solved the following discrete logarithm problem:

$$4^y \equiv 9 (mod\ 11). \tag{18}$$

Obtained QUBO problem consisted of 18 logical qubits. The problem has been embedded in the D-Wave Advantage computer, in the Pegasus topology, using 36 physical qubits.

Unfortunately, we could not quantumly solve discrete logarithm problems over larger fields using an efficient approach.

4.2 Solving discrete logarithm problem over $\mathbb{F}_{23}$ using mixed approach

However, in the case of the prime field $\mathbb{F}_{23}$, the smallest QUBO problem can be obtained using the brutal approach, we wanted to show the application of a mixed approach for solving discrete logarithm problem.

In this experiment, we solved the discrete logarithm problem over a field $\mathbb{F}_{23}$ which is the 5-bit prime field. The given generator was 2, and the order of 2 in $\mathbb{F}_{23}$ is 11. We solved the following discrete logarithm problem (with $m = 4$ and $k = 3$):

$$2^y \equiv 13 (mod\ 23). \tag{19}$$

Obtained QUBO problem consisted of 32 logical qubits. The problem has been embedded in the D-Wave Advantage computer, in the Pegasus topology, using 75 physical qubits.

Unfortunately, we were not able to quantumly solve discrete logarithm problem over $\mathbb{F}_{59}$ using a mixed approach. In such a case, the obtained QUBO problem consisted of 41 logical qubits, but after embedding in the D-Wave Advantage computer, the final problem required 130 physical qubits. We made in such a case several experiments, but we did not obtain the proper minimal energy.

4.3 Solving discrete logarithm problem over $\mathbb{F}_{59}$ using brutal approach

In the last experiment, we solved the discrete logarithm problem over a field $\mathbb{F}_{59}$ which is the 6-bit prime field. The given generator was 4, and the order of 4 in $\mathbb{F}_{59}$ is 29. We solved the following discrete logarithm problem:

$$4^y \equiv 27(mod\ 59). \tag{20}$$

Obtained QUBO problem consisted of 30 logical qubits. The problem has been embedded in the D-Wave Advantage computer, in the Pegasus topology, using 79 physical qubits.

4.4 Experiments summary

Unfortunately, we could not solve discrete logarithm problems over prime fields of a more significant length than 6 using quantum solver and D-Wave Advantage QPU.

We provide parameters of solution of discrete logarithm problem using D-Wave Advantage QPU over $\mathbb{F}_{11}$ using the efficient approach, $\mathbb{F}_{23}$ using the mixed approach, and $\mathbb{F}_{59}$ using the brutal approach.

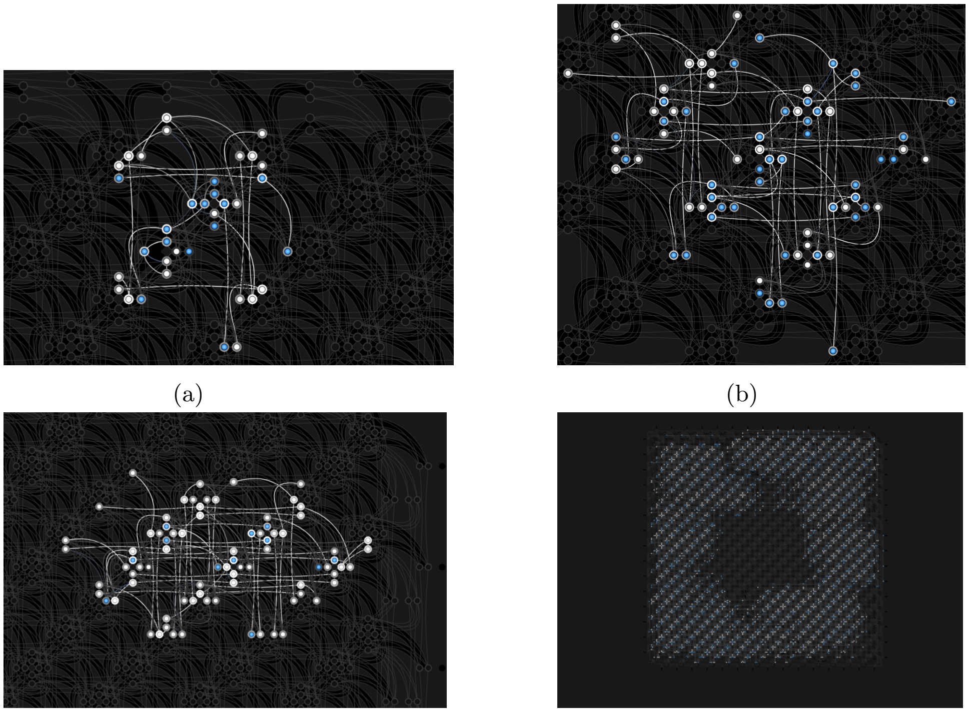
Parameter	Value
Name (chip ID)	Advantage_system4.1
Qubits	5,760
Topology	Pegasus
Number of reads	10,000
Annealing time (μs)	20
Anneal schedule	[[0,0],[20,1]]
H gain schedule	[[0,0],[20,1]]
Programming thermalization (μs)	1000

Table 1: D-Wave Advantage solver parameters used in solving QUBO problems equivalent to the discrete logarithm problems.

Figure 3 shows how different QUBO problems were embedded on the D-Wave Advantage computer.

Parameter	DLP over $\mathbb{F}_{11}$	DLP over $\mathbb{F}_{23}$	DLP over $\mathbb{F}_{59}$
Number of source variables	18	32	30
Number of target variables	36	75	81
Max chain length	4	5	4
Chain strength	8,258.58	20,000.00	2,025,065.28
QPU access time (μs)	959,248.6	1,324,651	913,450.6
QPU programming time (μs)	8,448.6	8,451	8,450.6
QPU sampling time (μs)	950,800	1,316,200	905,000
Total post processing time (μs)	11,209	17,384	17,023
Post processing overhead time (μs)	4,252	3,244	2,341

Table 2: Parameters used in solving QUBO problem equivalent to the problem of finding discrete logarithm over $\mathbb{F}_{11}$ using the efficient approach, $\mathbb{F}_{23}$ using the mixed approach and $\mathbb{F}_{59}$ using the brutal approach.



(a)

(b)

(c)

(d)

Fig. 3: Embedding of QUBO problems equivalent to discrete logarithm problems over the prime field $\mathbb{F}_{11}$ using efficient approach (Figure 3a), prime field $\mathbb{F}_{23}$ using mixed approach (Figure 3b), prime field $\mathbb{F}_{59}$ using brutal approach (Figure 3c), and prime field $\mathbb{F}_{65267}$ using mixed approach (Figure 3d).

We also checked the limitations of proposed methods using the D-Wave Advantage computer. We prepared a QUBO problem equivalent to a discrete logarithm problem over the 16-bit prime field $\mathbb{F}_{65267}$.

We used the mixed approach to try to solve the following discrete logarithm problem:

$$4^y \equiv 64643(mod\ 65267). \tag{21}$$

The equivalent QUBO problem consisted of 444 variables. We embedded this problem into the D-Wave Advantage QPU using 3,724 physical qubits. This embedding is presented in Figure 3d.

It is also worth noting that discrete logarithm problem over 17-bit prime, consisting of 537 logical qubits, could not be embedded into D-Wave Advantage QPU. According to our experiments, it looks that discrete logarithm problem over 16-bit prime field probably could not be properly solved. The biggest problem in such a case is a large number of variables and long chains, where maximal chain length is equal to 20. It is also interesting that discrete logarithm problem over 17-bit prime, consisting of 537 logical qubits, could not be embedded into D-Wave Advantage QPU.

The other observation is that using the brutal approach and classical CPU solver, we solved the QUBO problem equivalent to a discrete logarithm problem over a 10-bit prime in just 4.274 s. It is very interesting because this QUBO problem consists of 521 variables.

It looks that in the case of solving the QUBO problem, very important is the problem definition. The number of variables is also significant, but the number of connections between variables is often more critical. What is more, the QUBO problem obtained using a brutal approach and classical CPU solver may be easily solved even for the 521 variables problem. Unfortunately, the brutal approach is naturally limited by an exponential number of variables required to construct an equivalent QUBO problem.

5 Further works and conclusion

In this paper, we presented methods of transformation of discrete logarithm problem over prime fields to the QUBO problem. We showed different approaches to such transformation. The best methods allow one to obtain equivalent QUBO problem using approximately $2n^2$ variables (logical qubits). It is worth noting that in the case of factorization, in general, known methods allow transforming factorization problem to the QUBO problem using approximately $\frac{n^2}{4}$ variables if n is bit length of integer N that one wants to factorize.

The main result of the paper is a practical experiment, where discrete logarithm problem over $\mathbb{F}_{59}$ has been solved using D-Wave Advantage QPU. Even though it is a small problem, according to our knowledge, no one has reported a solution of discrete logarithm problem over prime fields using quantum methods until now.

Because the expected asymptotic time of solving the QUBO problem, even knowing the number of variables, is now unknown, it is hard to estimate the time in which presented QUBO problems could be solved. There are some expectations that for n variables QUBO problem, the minimal energy using quantum annealing should be found in $O\left(e^{\sqrt{n}}\right)$ [8], but more research in this area should be done. What is more, the presented methods should not outperform Shor's polynomial-time algorithm for large prime fields.

More research should be done to solve more significant problems using our methods. What is more, it seems that the presented methods may also be applied using quantum superconducting computers. Such researches have been done in the case of factorization problem [7], and it seems that transformation of discrete logarithm problem to find the ground state of the Hamiltonian and to solve then such problem using a superconducting quantum computer will also be possible.

References

1. D-WAVE, T.Q.C.C.: The d-wave advantage system: An overview. Technical report (2020)
2. D-WAVE, T.Q.C.C.: Getting started with the d-wave system. User manual (2020)
3. Dridi, R., Alghassi, H.: Prime factorization using quantum annealing and computational algebraic geometry. Scientific reports **7**, 43048 (2017)
4. Greene, T.: Google reclaims quantum computer crown with 72 qubit processor (2018)
5. Intel: The future of quantum computing is counted in qubits (2018)
6. Jiang, S., Britt, K.A., McCaskey, A.J., Humble, T.S., Kais, S.: Quantum annealing for prime factorization. Scientific reports **8**(1), 1–9 (2018)
7. Karamlou, A.H., Simon, W.A., Katabarwa, A., Scholten, T.L., Peropadre, B., Cao, Y.: Analyzing the performance of variational quantum factoring on a superconducting quantum processor. npj Quantum Information **7**(1), 1–6 (2021)
8. Mukherjee, S., Chakrabarti, B.K.: Multivariable optimization: Quantum annealing and computation. The European Physical Journal Special Topics **224**(1), 17–24 (2015)
9. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. pp. 124–134. Ieee (1994)
10. Sparkes, M.: A new quantum leader? (2021)
11. Wang, B., Hu, F., Yao, H., Wang, C.: prime factorization algorithm based on parameter optimization of ising model. Scientific Reports **10**(1), 1–10 (2020)
12. Wroński, M.: Index calculus method for solving elliptic curve discrete logarithm problem using quantum annealing. In: International Conference on Computational Science. pp. 149–155. Springer (2021)