Solving discrete logarithm problem over prime fields using quantum annealing and $\frac{n^3}{2}$ logical qubits

Michał Wroński[0000−0002−8679−9399]

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

Abstract. Shor's quantum algorithm for integer factorization and discrete logarithm is one of the fundamental approaches in modern cryptology. The application of Shor's algorithm requires a general-purpose quantum computer. On the other hand, there are known methods of transformation of factorization problem to the QUBO problem and then solving it using quantum annealing computing with approximately $\frac{n^2}{4}$ logical qubits, for example, using D-Wave computer. It is believed that this approach also may be helpful, primarily until large general-purpose quantum computers will exist. Until now algorithm of similar efficiency for computing discrete logarithm over prime fields was unknown. In this paper, we present a method of reducing discrete logarithm problem to the QUBO problem, which requires approximately $\frac{n^3}{2}$ logical qubits. We also show how to apply quantum annealing to compute discrete logarithm modulo composite numbers, where a quantum annealing factorization algorithm may be used to reduce discrete logarithm modulo composite to several discrete logarithm problems modulo prime number.

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

1 Introduction

Developing by Shor, the quantum algorithm for factorization and discrete logarithm computation [11] was one of the essential researches in modern cryptology. Since that time, 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 D-Wave Advantage. 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 Systems computer [9]. Moreover, the newest D-Wave computers have much more physical qubits than general-purpose quantum computers. It seems that, in some cases, D-Wave

computers may be used to solve 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 and modulo composite numbers into the optimization problem. The discrete logarithm problem may be transformed into the QUBO (Quadratic Unconstrained Boolean Optimization) problem, where constraints are exchanged by penalties added to the objective function.

Shor's quantum algorithm for factorization, discrete logarithm problem, and elliptic curves discrete logarithm problem is efficient (it works in polynomial time). Present quantum computers (even the most powerful) could solve DLP defined on 35-bit prime field $\mathbb{F}_p$ at most. Using the transformation of DLP to the QUBO problem, DLP of the same size should also be possible to break using D-Wave defined on at most 35-bit prime.

Presented in the paper, our contributions are:

- transformation of DLP over prime fields to the QUBO problem using approximately 2^n variables (brutal approach), where n is bit-length of p,
- transformation of DLP over prime fields to the QUBO using approximately $\frac{n^3}{2}$ variables (efficient approach), where n is bit-length of p,
- transformation of DLP modulo composite numbers $N = p_1^{a_1} \cdot \cdots \cdot p_k^{a_k}$ to the QUBO problem using approximately $max\{\frac{n^2}{4}, \frac{n_1^3}{2}, \ldots, \frac{n_k^3}{2}\}$, where n is bit-length of N and $n_1, \ldots, n_k$ are bit-lengths of $p_1, \ldots, p_k$ respectively.

2 Quantum annealing and cryptography

The Shor quantum algorithm for factorization and discrete logarithm began the race for constructing a quantum computer, which would solve real-world cryptographic problems. Nowadays, the two approaches of quantum computing for cryptography are the most exciting.

The first approach is quantum annealing, which is 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. What is essential from the cryptological point of view, several cryptographical 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 [6] factorized integer $200,099$, which result was later beaten by Jiang et al. [9], and by Wang et al. [12], who factorized 20-bit integer $1,028,171$.

On the other hand, general-purpose quantum computers have limited resources. The most powerful Intel, IBM, and Google quantum computers have 49, 53, and 72 qubits, respectively [8], [10], [7]. 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 [3], has 5436 working qubits. This quantum annealer allows solving general problems (presented in Ising or QUBO form) with up to $1,000,000$ variables and dense problems up to $20,000$ variables. A detailed description of how the D-Wave computer works may be found in [4].

2.1 Quantum annealing and D-Wave computer [4]

Quantum annealing uses quantum physics to find low-energy states of a problem and, therefore, the optimal (or near-optimal) combination of elements.

In D-Wave Systems, the quantum annealing is processed as follows. At first, it is necessary to introduce Hamiltonian, a mathematical description of some physical system in terms of its energies. For any particular input state of the system, the Hamiltonian returns the energy for that state. Unfortunately, for most non-convex Hamiltonians, finding the minimum energy state is an NP-hard problem that classical computers cannot solve efficiently.

In quantum computing, a Hamiltonian is a function mapping certain states (eigenstates) to energies. The critical fact is that the system's energy is well defined (eigenenergy) only when the system is in an eigenstate of the Hamiltonian. For the system in any other state, its energy is uncertain. The collection of eigenstates with defined eigenenergies makes up the eigenspectrum.

For the D-Wave system, the Hamiltonian may be represented as

$$H_{Ising} = \frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right) + \frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right), \qquad (1)$$

where $\hat{\sigma}_{x,z}^{(i)}$ are Pauli matrices, operating on a qubit q_i, and h_i and $J_{i,j}$ are the qubit biases and coupling strengths (nonzero values of h_i and $J_{i,j}$ are limited to those available in the working graph).

It is worth noting that the Hamiltonian is the sum of two terms, the initial Hamiltonian and the final Hamiltonian.

In computers using quantum annealing, one begins in the lowest-energy eigenstate of the initial Hamiltonian. During annealing, one introduces the problem Hamiltonian. The problem Hamiltonian contains the biases and couplers, and then one reduces the initial Hamiltonian influence. When the annealing ends, one is in an eigenstate of the problem Hamiltonian. It means that one has stayed in the minimum energy state throughout the quantum annealing process in an ideal situation. Hence, in the end, one is in the minimum energy state of the problem Hamiltonian. It means that one has an answer to the problem he wants to solve. By the end of the anneal, each qubit is a classical object.

2.2 QUBO and Ising problems

Ising problem [4] In the Ising model, variables appearing in this problem are "spin up" and "spin down," which corresponds to values 1 and -1. Relationships

between the spins are correlations or anti-correlation, which are represented by couplings. Then, the objective function expressed as an Ising model has the following form:

$$E_{Ising}(s) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} s_i s_j, \tag{2}$$

where the linear coefficients corresponding to qubit biases are h_i, and the quadratic coefficients corresponding to coupling strengths are $J_{i,j}$.

QUBO problem [4] QUBO (Quadratic Unconstrained Boolean Optimization) [4] is a significant problem with many real-world applications. One can express the QUBO model by the following optimization problem:

$$\min_{x \in \{0,1\}^n} x^T Q x, \tag{3}$$

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 be also viewed as problem of minimizing the function

$$f(x) = \sum_{i} Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j. \tag{4}$$

Conversion between Ising and QUBO problem [5] For given QUBO problem one can obtain equivalent Ising problem by setting

$$\begin{aligned} h_i &= \tfrac{Q_{i,i}}{2} + \textstyle\sum_{j=1}^{n} \tfrac{Q_{i,j}}{4}, \\ J_{i,j} &= \tfrac{Q_{i,j}}{4}, \end{aligned} \tag{5}$$

for $i \in \{1, \ldots, n\}$ and all $i < j$.

On the other hand, any Ising problem may be reformulated to the QUBO problem by defining [5]

$$\begin{aligned} Q_{i,i} &= 2\left(h_i - \textstyle\sum_{j=1}^{n} J_{i,j}\right), \\ Q_{i,j} &= 4J_{i,j}, \end{aligned} \tag{6}$$

for $i \in \{1, \ldots, n\}$ and all $i < j$.

2.3 Quantum annealing and factorization problem

As presented in the introduction, quantum annealing may be used to solve the factorization problem. We will shortly describe how to transform the factorization problem into the QUBO problem. We use the description presented in [9].

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 $f(x_1, x_2, x_3, \ldots, x_{l_1+l_2-2})^2 = (N - pq)^2$. 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 [9], in the following way

$$x_i x_j x_l \to u_k x_l = x_l u_k + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k). \tag{7}$$

It means that

$$x_i x_j x_l = u_k x_l = x_l u_k + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k), \tag{8}$$

if $x_i x_j = u_k$ and

$$x_i x_j x_l < x_l u_k + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k), \tag{9}$$

if $x_i x_j \neq u_k$.

It results in that $x_i x_j x_l$ term may be transformed to quadratic form by replacing $x_i x_j$ with u_k plus a constraint, given by penalty term:

$$min(x_i x_j x_l) = min\left(x_l u_k + 2(x_i x_j - 2u_k(x_i + x_j) + 3u_k)\right). \tag{10}$$

It is worth noting that for integer factorisation it is required $\binom{l_1-1}{2} + \binom{l_2-1}{2}$ auxiliary variables to form a quadratic cost function. If $l_1 = l_2$, then number of auxiliary variables is equal to $(l-1)(l-2)$ and $L = l(l-1)$ variables in total, what is approximately equal to $\frac{n^2}{4}$, where n is the bit-length 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.

2.4 Solving modular equations

We will transform the modular equations to the QUBO. We will show how it works considering linear modular equations.

Let's consider equation

$$ax \equiv b(mod\ p). \tag{11}$$

Because $x \in \{0, \ldots, p-1\}$, we can rewrite the Equation (11) as

$$a\left(u_1 + 2u_2 + \cdots + 2^{n-2}u_{n-1} + (p - 2^{n-1} + 1)u_n\right) \equiv b(mod\ p), \tag{12}$$

because $x = u_1 + 2u_2 + \cdots + 2^{n-2}u_{n-1} + (p - 2^{n-1})u_n$ for binary variables $u_1, \ldots, u_n$, and therefore $x \in \{0, \ldots, p-1\}$.

Now one should rewrite this equation as

$$\begin{aligned} &u_1(a\ mod\ p) + u_1(2a\ mod\ p) + \cdots + u_{n-1}(2^{n-2}a\ mod\ p) \\ &+u_n((p - 2^{n-1} + 1)a\ mod\ p) + (-b\ mod\ p) - kp = 0, \end{aligned} \tag{13}$$

where k is some integer. Let's note, that after such reduction, all monomials appearing in the equation above (instead of $-kp$) are positive. It means that one can bound k, because $(a \ mod \ p)+((-b) \ mod \ p)+(2a \ mod \ p)+\cdots+(2^{n-2}a \ mod \ p)+((p-2^{n-1}+1)a \ mod \ p) \geq kp$. What's more we can find general bound on k, because every monomial coefficient is from the set $\{0,\ldots,p-1\}$. Because we have $n+1$ monomials, we can find that $(p-1)(n+1) \geq kp$, which means that $k \leq \frac{(n+1)(p-1)}{p} < n+1$ and finally k may be written using $l = \lfloor \log_2(n+1) \rfloor + 1$ binary variables, similarly as x was written before. This idea may be found for example in [2].

Now we can rewrite the equation above

$$\begin{aligned} &f = u_1(a \ mod \ p) + u_1(2a \ mod \ p) + \cdots + u_{n-1}(2^{n-2}a \ mod \ p) \\ &+u_n((p-2^{n-1}+1)a \ mod \ p) + (-b \ mod \ p) - (k_1 + 2k_2 + \cdots + k_{l-1}(2^{l-2}) \\ &+k_l(n-2^{l-1}+1))p = 0. \end{aligned} \tag{14}$$

Finally, one should find the minimal energy of f^2 (this energy should be equal to 0), where f^2 will be indeed in the QUBO form.

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

We begin by defining discrete logarithm problem

$$g^y = h, \tag{15}$$

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

$$g^y \equiv h (mod \ p), \tag{16}$$

for integers $g, h \in \{1,\ldots,p-1\}, y \in \mathbb{Z}$.

Let m be the bit-length of $Ord(g)$. We begin by making following transformation. Let's note that if $y = 2^{m-1}u_m + \cdots + 2u_2 + u_1$, where $u_1,\ldots,u_m$ are binary variables, then

$$g^y = g^{2^{m-1}u_m+\cdots+2u_2+u_1} = g^{2^{m-1}u_m} \ldots g^{2u_2} g^{u_1}, \tag{17}$$

Let's also note that

$$g^{2^{i-1}u_i} = \begin{cases} 1, u_i = 0, \\ g^{2^{i-1}}, u_i = 1, \end{cases} \tag{18}$$

which is equivalent to

$$g^{2^{i-1}u_i} = 1 + u_i\left(g^{2^{i-1}} - 1\right). \tag{19}$$

Then holds

$$\begin{aligned} g^y &= \left(1 + u_m \left(g^{2^{m-1}} - 1\right)\right) \cdot \cdots \cdot \left(1 + u_2 \left(g^2 - 1\right)\right) \left(1 + u_1 \left(g - 1\right)\right) \\ &= \left(1 + u_m \left((g^{2^{m-1}} - 1) mod\ p\right)\right) \ldots \left(1 + u_2 \left((g^2 - 1) mod\ p\right)\right) \\ &\cdot \left(1 + u_1 \left((g - 1) mod\ p\right)\right). \end{aligned} \tag{20}$$

We can see that g^y may 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 $v_1 = u_1u_2$. 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 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

$$F = ((f - h)\ mod\ p - kp)^2 = 0, \tag{21}$$

where $k \in \mathbb{Z}$ and for every polynomial f, operation $f\ mod\ p$ is equivalent to the reduction all of coefficients of polynomial f modulo p. Let's note that k is bounded by the maximal number of monomials appearing in the polynomial $(f - h)\ mod\ p$, which is equal to 2^n. Finally, $k_{max} \leq \lfloor \frac{(2^n)(p-1)}{p} \rfloor < 2^n$ and bitlength of k is equal to $bl(k) = n$ at most. Moreover, we have to add penalties to the function F, according to Equation (7), obtaining $F_{Pen} = F + Pen$, where Pen are penalties added to the function, obtained during linearization. Polynomial F_{Pen} has minimal energy equal to 0 (our QUBO problem is constructed so that minimal energy, with high probability, is equal to 0 because in our QUBO problem appears constant energy offset).

An example of the application of the brutal approach is presented in Example 3.2.

3.1 Complexity of solving DLP as QUBO problem

If $Ord(g) = p - 1$, then the DLP transformation to the QUBO problem requires $2^n + n$ variables for n-bit prime p. Let us note that this exponential growth makes that the presented method may be applied only for small fields $\mathbb{F}_p$.

3.2 Example of application of brutal approach

Let's consider following discrete logarithm problem in field $\mathbb{F}_{23}$, equivalent to

$$2^y \equiv 6(mod\ 23). \tag{22}$$

We want to solve this problem and find y. At first, let's note that $Ord(2) = 11$ in field $\mathbb{F}_{23}$, so bit-length of n is equal to 4 at most.

Let' note that

$$\begin{aligned} f &= 2^y = 2^{u_4 2^3 + u_3 2^2 + u_2 2 + u_1} \\ &= (1 + u_4(2^{2^3} - 1))(1 + u_3(2^{2^2} - 1))(1 + u_2(2^2 - 1))(1 + u_1(2 - 1)) \\ &= (1 + 255u_4)(1 + 15u_3)(1 + 3u_2)(1 + u_1). \end{aligned} \tag{23}$$

Polynomial f is therefore equal to
$21u_1u_2u_3u_4 + 22u_1u_2u_3 + 6u_1u_2u_4 + 7u_1u_3u_4 + 21u_2u_3u_4 + 3u_1u_2 + 15u_1u_3 + 2u_1u_4 + 22u_2u_3 + 6u_2u_4 + 7u_3u_4 + u_1 + 3u_2 + 15u_3 + 2u_4 + 1$.

Then, one has to linearize f, by making a substitutions, which gives in result polynomial with 16 terms
$f_{lin} u_1 + 3u_2 + 15u_3 + 2u_4 + 3u_9 + 15u_{10} + 2u_{11} + 22u_{12} + 6u_{13} + 7u_{14} + 22u_{15} + 6u_{16} + 7u_{17} + 21u_{18} + 21u_{19} + 1$.

It is necessary to make 11 substitutions

$u_9 = u_1u_2, u_{10} = u_1u_3, u_{11} = u_1u_4, u_{12} = u_2u_3, u_{13} = u_2u_4, u_{14} = u_3u_4, u_{15} = u_3u_9, u_{16} = u_4u_9, u_{17} = u_4u_{10}, u_{18} = u_4u_{12}, u_{19} = u_9u_{14}$.

Now one computes $F = ((f_{lin} - h) \, mod \, p - kp)^2$, which gives in result polynomial consisted with 214 terms.

Moreover, we have to add penalties to the function F, according to Equation (7), obtaining $F_{Pen} = F + Pen$, where Pen are penalties added to the function, obtained during linearization. Finally, the objective function F_{Pen} is a quadratic function of 19 variables and 191 terms and is in the QUBO form.

Now one can solve this problem using, for example, a quantum annealing computer, which gives in result
$u_1 = 1, u_2 = 0, u_3 = 0, u_4 = 1, u_5 = 0, u_6 = 0, u_7 = 0, u_8 = 0, u_9 = 0, u_{10} = 0, u_{11} = 1, u_{12} = 0, u_{13} = 0, u_{14} = 0, u_{15} = 0, u_{16} = 0, u_{17} = 0, u_{18} = 0, u_{19} = 0$,
which has minimal energy equal to 0 (our QUBO problem is constructed in such a way that minimal energy, with high probability, is equal to 0 because in our QUBO problem appears constant energy offset).

Finally, $y = 8u_4 + 4u_3 + 2u_2 + u_1 = 8 + 1 = 0$ and, indeed, $2^9 = 6(mod \, 23)$.

4 Tranformation of discrete logarithm problem to the QUBO problem - efficient approach

Let's consider as before that $x_i = g^{u_i 2^{i-1}} = 1 + u_i \left(g^{2^{i-1}} - 1 \right)$, for $i = \overline{1, m}$.

Now the DLP problem, given by Equation (15) may be transformed to the problem of finding solution of

$$x_1 x_2 \cdots x_m \equiv h(mod \, p). \tag{24}$$

As we know from the brutal approach, if one multiplies $x_1 x_2 \cdots x_m$ and then tries to make linearization, then the whole problem will require $2^m + m$ variables in total.

We will decrease the number of variables in the following way, which is described in [2], in the context of quadratization of polynomials of degree greater than 2 over finite fields. At first, let's note that for each pair $x_1x_2, x_3x_4, ...$ we can create new variable $v_i = x_ix_{i+1}$, which is equivalent to $v_i \equiv x_ix_{i+1}(mod\ p)$. 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 $v_{m-3}v_{m-2} \equiv h(mod\ p)$, so the number of auxiliary variables v_i is equal to $m-2$.

The scheme of such a tree for $m = 8$ is presented in Figure 1.

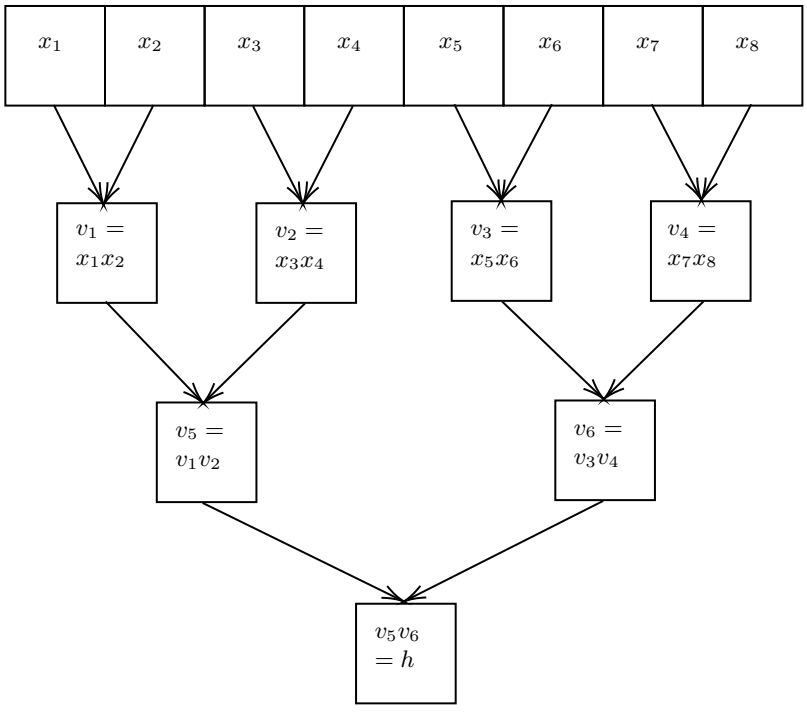

Fig. 1. An example of creation of auxiliary variables for $m = 8$.

What's more, each of equations for $u_1, u_2, u_{n-2}, \ldots, u_{n-3}, u_{n-2}, u_{n-3}u_{n-2}$ need to be transformed to the equation over integers.

$$\begin{cases} f_1 = (v_1 - x_1x_2) \ mod \ p - k_1p = 0, \\ f_2 = (v_2 - x_3x_4) \ mod \ p - k_2p = 0, \\ \ldots \\ f_{m-3} = (v_{m-3} - v_{m-4}v_{m-5}) \ mod \ p - k_{m-3}p = 0, \\ f_{m-2} = (v_{m-2} - v_{m-6}v_{m-7}) \ mod \ p - k_{m-2}p = 0, \\ f_{m-1} = (v_{m-3}v_{m-2} - h) \ mod \ p - k_{m-1}p = 0, \end{cases} \tag{25}$$

We will precisily count how many auxiliary variables are necessary. Let's note that for variables $v_1, \ldots, v_{\lfloor \frac{m}{2} \rfloor}$ it is necessary n bits to represent v_i and only 3 bits to represent k_i, because $(v_i - x_jx_{j+1}) \ mod \ p \equiv v_i - 1 + \left(-\left(g^{2^{j-1}} + 1\right) \ mod \ p\right) u_j + \left(-\left(g^{2^j} + 1\right) \ mod \ p\right) u_{j+1} + \left(-\left((g^{2^{j-1}} + 1)(g^{2^j} + 1)\right) \ mod \ p\right) u_ju_{j+1} (mod \ p)$. It holds that $u_j, u_{j+1}, u_ju_{j+1} \leq p - 1$, as same as $v_i \leq p - 1$. Because each x_i is greater than 0, then $v_i - 1$ is nonnegative. It means that $(v_i - x_jx_{j+1}) \ mod \ p \leq 4(p - 1)$ and therefore $k_i \leq 3$.

For variables $v_{\lfloor \frac{m}{2} \rfloor + 1}, \ldots, v_{m-2}$ there are also necessary n bits to represent each of variable. Additionally, for every $i = \lfloor \frac{m}{2} \rfloor + 1, \ldots, m - 2$, the polynomial $(v_jv_{j+1}) \ mod \ p$ consist of n^2 terms, with positive coefficients, where each is bounded by $p - 1$. It means that for $(v_i - v_jv_{j+1}) mod \ p$ holds that $k_i \leq \lfloor \frac{(n^2+1)(p-1)}{p} \rfloor < n^2 + 1$. Moroever, during linearization of $(v_i - v_jv_{j+1}) \ mod \ p$ it is necessary to linearize terms appearing in v_jv_{j+1}, which requires n^2 variables. Finally, for equation $(v_{m-3}v_{m-2} - h) \ mod \ p - k_{m-1}p$, the variable k_{m-1} is also bounded by $\lfloor \frac{(n^2+1)(p-1)}{p} \rfloor \leq n^2 < n^2 + 1$. It means that the bit-length of each of variables $k_{\lfloor \frac{m}{2} \rfloor + 1}, \ldots, k_{m-2}$ is bounded by $\lfloor \log_2 n^2 \rfloor + 1$.

Because for each variable holds that $v_i \in \{1, \ldots, p - 1\}$, it means that such variable requires n bits. Let's note that each polynomial $(v_k - x_ix_{i+1}) \ mod \ p$ has all coefficients positive, as same as polynomial $(v_{m-3}v_{m-2} - h) \ mod \ p$.

Let's note that for $i = 1, \ldots, \lfloor \frac{m}{2} \rfloor$ linearization of $f_1, \ldots, f_{\lfloor \frac{m}{2} \rfloor}$ requires only linearization of x_ix_{i+1} terms, which requires only 1 auxiliary variable for each equation. For all other equations for $i = f_{\lfloor \frac{m}{2} \rfloor} + 1, \ldots, m - 1$ the linearization of f_i requires the linearization of v_jv_k. Because variables v_j, v_k require n bits each, the linearization of v_jv_k requires n^2 auxiliary variables.

Let's 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

$$F_{Pen} = (f_{lin_1})^2 + \cdots + (f_{lin_{m-1}})^2 + Pen, \tag{26}$$

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

Finally, obtained QUBO problem requires $\left(-\lfloor \frac{m}{2} \rfloor + m - 1\right)\left(n^2 + \lfloor \log_2 \left(n^2\right) \rfloor + 1\right) + 3\lfloor \frac{m}{2} \rfloor + (m-2)n$, which is approximately equal to $\frac{mn^2}{2}$ 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 $\frac{n^3}{2}$.

4.1 Example of application of efficient approach

Let's consider following discrete logarithm problem in field $\mathbb{F}_{23}$

$$2^y \equiv 6 (mod\ 23), \tag{27}$$

which means that $g = 2, h = 6$.

As previous, let's define $x_1 = 1 + u_1\left((g-1) mod\ p\right) = 1 + u_1, x_2 = 1 + u_2\left((g^2-1) mod\ p\right) = 1 + 3u_2, x_3 = 1 + u_3\left((g^4-1) mod\ p\right) = 1 + 15u_3, x_4 = 1 + u_4\left((g^8-1) mod\ p\right) = 1 + 2u_4$. Now let's define

$$\begin{cases} v_1 \equiv x_1 x_2 (mod\ 23), \\ v_2 \equiv x_3 x_4 (mod\ 23), \\ v_1 v_2 \equiv 6 (mod\ 23), \end{cases} \tag{28}$$

where $v_1 = u_5 + 2u_6 + 4u_7 + 8u_8 + 7u_9$, and $v_2 = u_{10} + 2u_{11} + 4u_{12} + 8u_{13} + 7u_{14}$. Let's note that

1. $(v_1 - x_1 x_2) \mod 23 - 23k_1 = 0$ is equal to $f_1 = 20u_1u_2 + 22u_1 + 20u_2 + u_5 + 2u_6 + 4u_7 + 8u_8 + 7u_9 - 23u_{15} - 46u_{16} - 1$,
2. $(v_2 - x_3 x_4) \mod 23 - 23k_2 = 0$ is equal to $f_2 = 16u_3u_4 + 8u_3 + 21u_4 + u_{10} + 2u_{11} + 4u_{12} + 8u_{13} + 7u_{14} - 23u_{17} - 46u_{18} - 1$,
3. $(v_1 v_2 - 6) \mod 23 - 23k_3 = 0$ is equal to $f_3 = u_5u_{10} + 2u_5u_{11} + 4u_5u_{12} + 8u_5u_{13} + 7u_5u_{14} + 2u_6u_{10} + 4u_6u_{11} + 8u_6u_{12} + 16u_6u_{13} + 14u_6u_{14} + 4u_7u_{10} + 8u_7u_{11} + 16u_7u_{12} + 9u_7u_{13} + 5u_7u_{14} + 8u_8u_{10} + 16u_8u_{11} + 9u_8u_{12} + 18u_8u_{13} + 10u_8u_{14} + 7u_9u_{10} + 14u_9u_{11} + 5u_9u_{12} + 10u_9u_{13} + 3u_9u_{14} - 23u_{19} - 46u_{20} - 92u_{21} - 184u_{22} - 161u_{23} + 17$,

where $k_1 = u_{15} + 2u_{16}, k_2 = u_{17} + 2u_{18}, k_3 = u_{19} + 2u_{20} + 4u_{21} + 8u_{22} + 7u_{23}$.

Finally, polynomial F_{Pen} is equal to $F_{Pen} = (f_{lin_1})^2 + (f_{lin_2})^2 + (f_{lin_3})^2 + Pen$, where Pen denotes penalties obtained during linearization. What is more, the polynomial F_{Pen} is a polynomial consisted of 524 terms.

The final polynomial (after quadratization and addition of penalties) is polynomial with 551 terms.

Finally, one can solve this problem using, for example, a quantum annealing computer, which gives in result
$u_1 = 1, u_2 = 0, u_3 = 0, u_4 = 1, u_5 = 0, u_6 = 1, u_7 = 0, u_8 = 0, u_9 = 0, u_{10} = 1, u_{11} = 1, u_{12} = 0, u_{13} = 0, u_{14} = 0, u_{15} = 1, u_{16} = 0, u_{17} = 1, u_{18} = 0, u_{19} = 1, u_{20} = 0, u_{21} = 0, u_{22} = 0, u_{23} = 0, u_{24} = 0, u_{25} = 0, u_{26} = 0, u_{27} = 0, u_{28} = 0, u_{29} = 0, u_{30} = 0, u_{31} = 0, u_{32} = 0, u_{33} = 0, u_{34} = 0, u_{35} = 0, u_{36} = 0, u_{37} = 0, u_{38} = 0, u_{39} = 0, u_{40} = 0, u_{41} = 0, u_{42} = 1, u_{43} = 1, u_{44} = 0, u_{45} = 0, u_{46} = 0, u_{47} = 0, u_{48} = 0, u_{49} = 0, u_{50} = 0,$
which has minimal energy equal to 0 (our QUBO problem is constructed so that minimal energy, with high probability, is equal to 0 because our QUBO problem appears constant energy offset).

Finally, $y = 8u_4 + 4u_3 + 2u_2 + u_1 = 8 + 1 = 0$ and, indeed, $2^9 = 6(mod\ 23)$.

5 Computing discrete logarithm problem modulo composite integers using quantum annealing

Let us define discrete logarithm problem

$$g^y = h, \tag{29}$$

in the ring $\mathbb{Z}_N$.

Let us suppose that N is composite integer and $N = p_1^{a_1} \cdot \cdots \cdot p_k^{a_k}$, where $p_1, \ldots, p_k$ are primes which are unknown yet.

According to Bach [1], it is a well-known fact that if one can fast compute factorization of m and then also fast compute discrete logarithm modulo prime, then discrete logarithm modulo composite number also may be computed fast.

If factorisation of n is known, then one can use Pohlig-Hellman reduction and reduce problem given by the Equation (29) to computation of discrete logarithms modulo prime: find such $y_1, \ldots, y_k$ that

$$\begin{cases} g^{y_1} \equiv h(mod\ p_1^{a_1}), \\ \ldots \\ g^{y_k} \equiv h(mod\ p_k^{a_k}). \end{cases} \tag{30}$$

Solving discrete logarithm over composite number, with application of Pohlig-Hellman reduction, requires then $O\left(F(N) + D(p_1^{a_1}) + \cdots + D(p_k^{a_k})\right)$ operations, where $F(N)$ is number of operations necessary for factorization of N and $D(p^a)$ is number of operations necessary to solve discrete logarithm over p^a. Let's note that also factorisation may be performed using quantum annealing approach and it should be made recursevily until full factorisation is found. What's more, the number of variables of equivalent QUBO problem is equal to $O\left(\frac{n^2}{4}\right)$, where n is bit-length of N. Because it is well known that computing of discrete logarithm modulo p^a may be reduced to computation of discrete logarithm modulo p, the

number of variables of equivalent QUBO problem necessary for solving discrete logarithm problem over $\mathbb{Z}_N$ is equal to approximately $max\left\{\frac{n^2}{4}, \frac{n_1^3}{2}, \ldots, \frac{n_k^3}{2}\right\}$, where n is bit-length of N and $n_1, \ldots, n_k$ are bit-lengths of $p_1, \ldots, p_k$ respectively.

6 Results, and discussion

It is worth noting that using D-Wave Advantage should solve the QUBO problem, which consists of $20,000$ variables, considering this problem as dense, and a problem which consists of up to $1,000,000$ variables considering this problem as general. It is worth noting that these are only theoretical expectations, and more research should be done in this field.

In brutal approach, the number of variables is approximately equal to 2^n, where n is the bit-length of p, so considering DLP as dense, it would be possible to break DLP over 16 bits prime field $\mathbb{F}_p$, and considering DLP as the general problem, it would be possible to break DLP over $19 - 20$ bits prime field $\mathbb{F}_p$.

In an efficient approach, the number of variables is approximately equal to $\frac{n^3}{2}$, where n is the bit-length of p, so considering DLP as dense, it would be possible to break DLP over 35 bits prime field $\mathbb{F}_p$, and considering DLP as the general problem (what seems to be more accurate in this case) it would be possible to break DLP over 127 bits prime field $\mathbb{F}_p$ at most.

On the other hand, the most powerful quantum computer nowadays, the IBM quantum computer, would solve (at most) DLP for 35 bits long prime, which requires about 71 qubits. Moreover, no DLP solution using a general-purpose quantum computer has been reported so far (as well as using a quantum annealing computer), which means that currently, available quantum computers have limited DLP computing capabilities.

Number of bits	Shor's DLP and factorization [qb]	Factorization using QUBO [qb]	DLP using QUBO efficient approach [qb]	DLP using QUBO brutal approach [qb]
n	$2n+1$	$\approx \frac{n^2}{4}$	$\approx \frac{n^3}{4}$	$\approx 2^n$
...	...	...	...	...
35	71	307	$21,438$	2^{35}
...	...	...	...	...
127	255	$4,033$	$1,024,192$	2^{127}

Table 1. The comparison of number of logical qubits required to solve different problems using quantum computing.

7 Conclusion and further works

The transformation of DLP to the QUBO problem is an alternative approach to Shor's algorithm to solve this problem using quantum computing. It is worth noting that in theorem, the most powerful general-purpose quantum computers would be able to break 35-bits DLP at most. Unfortunately, it would be tough to do it in practice. Transforming DLP to the QUBO problem and using quantum annealers for solving this problem nowadays seems to have more potential to solve real size problems. As presented in Table 1, D-Wave Advantage has the potential to break DLP up to 127 bits prime field $\mathbb{F}_p$ considering DLP as a general problem (not a dense one).

The efficiency of the approach using QUBO may be increased by applying the following improvements:

- applying the different algorithm of quadratization of the resulting polynomial, which would result in less number of total variables or smaller connectivity between variables,
- modifying a method of translation of discrete logarithm problem to the QUBO problem to obtain the larger value of minimal energy gap and thus decreasing the probability of obtaining suboptimal solution instead of the optimal solution,
- manual embedding of the given QUBO problem to the D-Wave Advantage computer.

It seems that if improvements above would be possible to apply, the solving of discrete logarithm problem using transformation to the QUBO problem would be much more efficient and could be solved on D-Wave for larger fields. It is an open problem if it is possible to transform DLP to the QUBO problem using approximately $\frac{n^2}{4}$ logical qubits as in the case of factorization, or even less. Summing up, this approach seems to have potential and more research in this area should be done.

References

1. Bach, E.: Discrete logarithms and factoring. Tech. Rep. UCB/CSD-84-186, EECS Department, University of California, Berkeley (Jun 1984), http://www2.eecs.berkeley.edu/Pubs/TechRpts/1984/5973.html
2. Chen, Y.A., Gao, X.S., Yuan, C.M.: Quantum algorithm for optimization and polynomial system solving over finite field and application to cryptanalysis. arXiv preprint arXiv:1802.03856 (2018)
3. D-WAVE, T.Q.C.C.: The d-wave advantage system: An overview. Technical report (2020)
4. D-WAVE, T.Q.C.C.: Getting started with the d-wave system. User manual (2020)
5. Djidjev, H.N., Chapuis, G., Hahn, G., Rizk, G.: Efficient combinatorial optimization using quantum annealing. arXiv preprint arXiv:1801.08653 (2018)
6. Dridi, R., Alghassi, H.: Prime factorization using quantum annealing and computational algebraic geometry. Scientific reports **7**, 43048 (2017)

7. Greene, T.: Google reclaims quantum computer crown with 72 qubit processor (2018)
8. Intel: The future of quantum computing is counted in qubits (2018)
9. 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)
10. Shankland, S.: Ibm's new 53-qubit quantum computer is its biggest yet (2019), https://www.cnet.com/news/ibm-new-53-qubit-quantum-computer-is-its-biggest-yet/
11. 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)
12. 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)