INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 1, PP. 55–60
Manuscript received December 2, 2024; revised March, 2025. doi: 10.24425/ijet.2025.153544

The generalized method of solving ECDLP using quantum annealing

Łukasz Dzierzkowski

***Abstract*—This paper presents a generalization of a method allowing the transformation of the Elliptic Curve Discrete Logarithm Problem (ECDLP) over prime fields to the Quadratic Unconstrained Binary Optimization (QUBO) problem. The original method requires that a given elliptic curve model has complete arithmetic. The new one has no such restriction, which is a breakthrough. Since the mentioned obstacle is no longer a problem, the latest version of the algorithm may be used for any elliptic curve model. As a result, one may use quantum annealing to solve ECDLP on any given model of elliptic curves.**

***Keywords*—cryptology; cryptanalysis; elliptic curves; discrete logarithm; ECDLP; quantum annealing**

I. Introduction

ELLIPTIC curves have many practical applications nowadays, such as: encryption, key exchange, digital signatures, primality testing and factorization. The first three of them require ensuring safe use, which means that any eavesdropper will not be able to recover protected data or forge a valid signature. One of the mechanisms, that allow protocols in the elliptic curve cryptography (ECC) to remain secure, is the hardness of solving the elliptic curve discrete logarithm problem (ECDLP). One can choose one of two possible ways to deal with this problem: classical or quantum.

While considering the first option, many algorithms are available. The most desired are those, which may be used in arbitrary cases, e.g.:

- the method of Pohlig-Hellman [1],
- Baby step, giant step method [2], [3],
- Pollard's ρ method [4],
- Pollard's λ method [4].

They all have one thing in common – fully exponential computational complexity in a general case [5]. That means, ECDLP used in modern cryptosystems cannot be solved in a reasonable time with a classical algorithm. Moreover, it is hard to notice any prospect of changing this situation, at least in the near future. If so, other ways should be considered.

The second option is quantum computing. The most known device in the described category is a general purpose quantum computer (GPQC). There is only one known algorithm, which may be used to solve ECDLP on the mentioned device – Shor's algorithm. However, none of the ECDLP examples have been solved with this method so far. The reason for this state of affairs was mentioned by Martin Roetteler et al. in [6]. The authors of that paper showed, what is the number of Toffoli gates required to solve ECDLP with Shor's algorithm [7]. For an elliptic curve on some n-bit prime field, it could be even $448n^3 \log_2(n) + 4090n^3$, which exceeds the achievable resources.

Fortunately, quantum computing does not end with GPQC. In the field of cryptography, quantum annealing (QA) is increasingly used. So far, many applications of QA in cryptanalysis have been described, i.e.:

- solving discrete logarithm problem (DLP) [8],
- conducting an algebraic attacks on block ciphers [9], [10],
- conducting an algebraic attack on stream ciphers [11],
- integer factorization [12]–[15],
- **solving ECDLP** [16]–[18].

The last item on the list concerns the problem that is the main topic of this paper. As may be seen, solving ECDLP with QA has already been described. Nevertheless, none of the mentioned articles presents a fully-quantum method of solving the ECDLP for an arbitrary case ([16] uses both classical and quantum computations and [17], [18] require to conduct the operations on an elliptic curve with complete arithmetic).

The breakthrough will be presented in this article. The method described below does not require any computations on a classical computer (just converting the ECDLP to some specific form) and may be used for any elliptic curve model, even without complete arithmetic.

II. Theory

The quantum annealer D-Wave, which allows building applications to access and use QA in the cloud, accepts a few different classes of problems, for example:

1) Binary Quadratic Models (BQC),
2) Constrained Quadratic Models (CQM),
3) Discrete Quadratic Models (DQM).

Solving ECDLP with QA requires transforming the problem to the BQM. There are two possible representations in that class: the Ising model and the quadratic unconstrained binary optimization model (QUBO). Switching between models is almost effortless. Due to practical purposes and because of the differences in the way of representing variables in both

This work was supported by the grant No. DOB/002/RON/ID1/2018 by The National Centre for Research and Development.

Ł. Dzierzkowski is with Faculty of Cybernetics, Military University of Technology, Warsaw, Poland (e-mail: lukasz.dzierzkowski[at]wat.edu.pl).

models, the QUBO model is more suitable for solving ECDLP and therefore it was used.

In the below subsections, brief descriptions of ECDLP and QUBO are provided. Then the original way of solving the described problem is recalled. Eventually, in the last paragraph of this chapter, the idea for generalizing and improving the method is described.

A. ECDLP

Elliptic curve discrete logarithm problem is defined as finding such integer y, that for given elliptic curve E over a prime field $\mathbb{F}_p$

$$[y]P = \underbrace{P + P + \cdots + P}_{\text{y addends}} = Q, \tag{1}$$

where P and Q are points on the curve E and $y \in \{1, \ldots, Ord(P) - 1\}$. Let m be the bitlength of $Ord(P)$, then y may be written with m binary variables u_i as

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

what makes possible to transform Eq. (1) into

$$\begin{aligned} Q = [y]P = &\ [2^{m-1}u_m + \cdots + 2u_2 + u_1]P \\ = &\ [2^{m-1}u_m]P + \ldots + [2u_2]P + [u_1]P \\ = &\ [u_m]([2^{m-1}]P) + \ldots + [u_2]([2]P) + [u_1]P \\ = &\ P_m + \ldots + P_2 + P_1. \end{aligned} \tag{3}$$

As an example, curve $E : y^2 = x^3 - 3x + 63$ over a prime field $\mathbb{F}_{1021}$ with order $ord(E) = 964$ may be given. For points $P = (74, 841)$ and $Q = [k]P = (1017, 824)$, solving ECDLP means finding the proper k value. For a finite field of a 10-bit length, dealing with this problem manually may be tedious, but any computer can accomplish it in fractions of a second and calculate the solution, which is $k = 43$. The situation is quite different in modern cryptosystems, where the bit length of a finite field is expressed using hundreds of bits.

B. QUBO

Quadratic unconstrained binary optimization model is presented as

$$\min_{x \in \{0,1\}^N} x^T Q x, \tag{4}$$

where Q is an $N \times N$ upper-diagonal matrix of real weights, and x is a vector of binary variables. It can be also defined as minimizing the function

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j. \tag{5}$$

As an example, a function $f(x_1, x_2, x_3) = 3x_1^2 - 3x_3^2 + 5x_2x_3$ may be given. With the vector x and the matrix Q, it may be written as

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \; Q = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 5 \\ 0 & 0 & -3 \end{bmatrix}.$$

Minimal energy of the given function is -3 for $(x_1, x_2, x_3) = (0, 0, 1)$.

C. Transformation

Since the starting and ending points for solving ECDLP have been described above, the last missing part is the road in between. It was minutely characterized in [17]. Below one can find a briefer explanation.

A single summand from the third line of Eq. (3) can be written as

$$[u_i]\left([2^{i-1}]P\right) = \begin{cases} \mathcal{O} & \text{for } u_i = 0, \\ [2^{i-1}]P & \text{for } u_i = 1. \end{cases} \tag{6}$$

The two cases can be united into one expression as

$$[u_i]\left([2^{i-1}]P\right) = \mathcal{O} + u_i\left([2^{i-1}]P - \mathcal{O}\right) \tag{7}$$

and for the corresponding values of u_i from Eq. (6), the results are as assumed above. If the arithmetic of points on the chosen elliptic curve model is complete, the above formula may be divided into affine coordinates

$$\begin{cases} P_{i,x} = \mathcal{O}_x + u_i\left([2^{i-1}]P_x - \mathcal{O}_x\right), \\ P_{i,y} = \mathcal{O}_y + u_i\left([2^{i-1}]P_y - \mathcal{O}_y\right). \end{cases} \tag{8}$$

Based on these equations, every precomputed multiplicity (only powers of 2) of point P from Eq. (3) can be represented. In order to improve the intuition of the whole process, an illustration may be helpful.

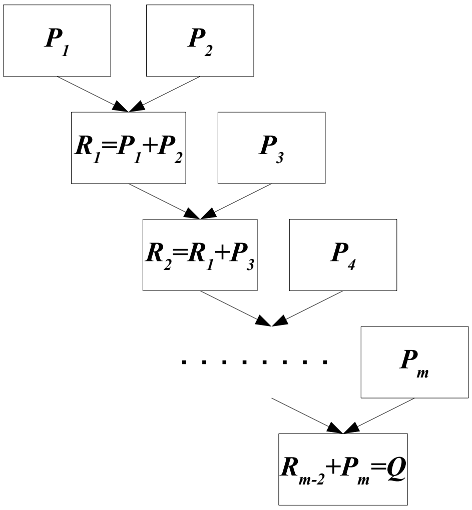

Fig. 1. The decomposition of ECDLP [17, Fig. 2]

The way of representing points P_i is described above and the form of a given point Q is with the given coordinates, just like in the example at the end of subsection II-A. The only one unknown so far is the idea of representing the sum points R_i. Since ECDLP is given over some finite field, the coordinates of points R_i must belong to this field. If so, each of them may be written with n binary variables, where n is the bit length of a field characteristic p.

Having knowledge about the representation forms of the points P_i, R_i and Q, one may insert them into complete arithmetic equations on the selected elliptic curve model and perform the whole process. Well described example on the Edwards curve may be found in [17]. Moreover, in continuation of that research the authors have noticed, that changing the representation of a sought multiplicity of a point P from binary to, for example, ternary may improve the process and lower the number of necessary resources. The results may be found in [18].

However, there is still a problem: how to solve the ECDLP which is defined on curve models without complete arithmetic?

D. Problems with the original method

First of all, before the generalization is described, one needs to think about the motivation and ideas of how to go beyond the restrictions. The following conditions for solving ECDLP on the particular elliptic curve model have been mentioned by the authors of [17] for using the method from subsection II-C:

1) arithmetic with a small number of multiplications,
2) neutral element represented by affine coordinates,
3) complete arithmetic.

The first condition seems to affect the efficiency of the computations, so it may be omitted at the expense of increasing the number of necessary resources.

The second prerequisite must be satisfied in the part of the existence of the numerical representation of the neutral element $\mathcal{O}$. Without that, points P_i cannot be represented with the form from Eq. (7) and thus with Eqs. (8). The type of coordinates influences mainly the efficiency, as, for example, computations in affine coordinates require less resources than in projective ones.

The last requirement forces using such arithmetic, that works correctly for all inputs. That means, it does not matter if the arguments are two different points, two same points or some point and the neutral point, for all these cases the formula should proceed and return a correct result. Due to the form of ECDLP decomposition in Eq. (3), the arithmetic does not have to be unified (the same for adding and doubling points), because doubling points never happens, so adding one is enough. However, for the proper algorithm to work, the formula must take into account adding an arbitrary point with a neutral point $\mathcal{O}$.

Summing up, the biggest problem in generalizing the original method is the behavior of the algorithm when facing the neutral point. For elliptic curve models, in which the neutral point does not have a numerical form or the arithmetic does not cooperate with the neutral point (like in short Weierstrass model), the original method from [17] cannot work.

E. Generalization - introduction

To face that problem, the algorithm must be modified. The idea is to perform operations one level higher than in the original method. This means, that instead of converting points P_i like in Eq. (7) and then inserting it into the addition formula, one may use the binary variables u_i from Eq. (2) and the third line of Eq. (3) and consider cases in addition $[u_i]P_i + [u_j]P_j$.

The first one, when both u_i and u_j are positive. If so, the correct value is a point $P_i + P_j$.

The second one, when only u_i is positive. Then, the proper result is P_i.

The third one, when only u_j is positive. The correct result is P_j.

The last one, when neither u_i nor u_j is positive. Thinking logically, this is a case for a neutral point $\mathcal{O}$. However, it should not be used, because there may be an elliptic curve model, on which the corresponding arithmetic does not cooperate with the neutral points. For now, this case is unresolved but it will be below.

Based on the above considerations, the generalized addition may be written as

$$[u_i]P_i + [u_j]P_j = \begin{cases} ??? & \text{for } u_i = 0 \wedge u_j = 0, \\ P_i & \text{for } u_i = 1 \wedge u_j = 0, \\ P_j & \text{for } u_i = 0 \wedge u_j = 1, \\ P_i + P_j & \text{for } u_i = 1 \wedge u_j = 1. \end{cases} \tag{9}$$

As long as variables u_i and u_j are binary, the above cases may be transformed with boolean algebra into the expression

$$\begin{aligned}[u_i]P_i + [u_j]P_j = \; & ??? + [u_i(1-u_j)]P_i + [(1-u_i)u_j]P_j \\ & + [u_i \cdot u_j](P_i + P_j). \end{aligned} \tag{10}$$

Now the "???" signs have to be replaced with a numerical value. It can be done in at least two ways. Before the further description is presented, a few things should be noticed. The illustration in Fig. 1 will be helpful.

Let's think about the first addition

$$P_1 + P_2 = [u_1]P + [u_2]\left([2]P\right).$$

If at least one out of the variables u_i or u_j is positive, the problematic case does not appear. What is more, that incident will not occur during subsequent additions, as all of the first arguments, being the results of all previous operations, will be different than the neutral element. Since binary variables in the first addition are the less significant bits of a multiplicity y, the analyzed "???" case appears when y is divisible by 4.

F. Generalization - first idea

The first idea of dealing with the problem is very simple. While solving ECDLP $[y]P = Q$, points P, Q and the elliptic curve E are known. If so, one may perform an addition $Q + P = [y+1]P$. Since the result of finding y is not correct, it may indicate that y is divisible by 4. If so, $y + 1$ does not have this property so the algorithm should solve the problem. The small inconvenience is that the result must be reduced by 1. While using this solution, the case with u_i and u_j both negative can be omitted, so it is equal

$$[u_i]P_i + [u_j]P_j = \begin{cases} P_i & \text{for } u_i = 1 \wedge u_j = 0, \\ P_j & \text{for } u_i = 0 \wedge u_j = 1, \\ P_i + P_j & \text{for } u_i = 1 \wedge u_j = 1, \end{cases} \tag{11}$$

and thus

$$\begin{aligned}[u_i]P_i + [u_j]P_j = \; & [u_i(1-u_j)]P_i + [(1-u_i)u_j]P_j \\ & + [u_i \cdot u_j](P_i + P_j). \end{aligned} \tag{12}$$

G. Generalization - second idea

The second idea is to conditionally add point P during the first addition. Thanks to this, the result of a first operation will always be different than $\mathcal{O}$ and at the same time, potential problems in subsequent additions are eliminated. However, adding extra P makes, that the solver has to deal with a problem $[y+1]P = [y]P$, which is not true if $P \neq \mathcal{O}$. That is why an extra P has to be added to Q as well. The modification was presented in the Fig. 2.

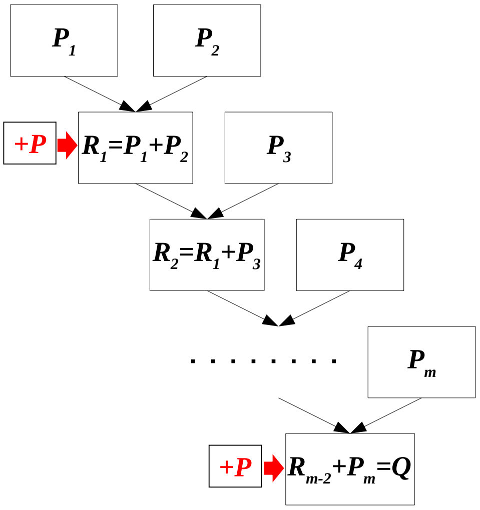

Fig. 2. The modified decomposition of ECDLP

Now the final version of arithmetic formulas may be presented. The first addition should be done with

$$[u_i]P_i + [u_j]P_j = \begin{cases} P & \text{for } u_i = 0 \wedge u_j = 0, \\ P_i & \text{for } u_i = 1 \wedge u_j = 0, \\ P_j & \text{for } u_i = 0 \wedge u_j = 1, \\ P_i + P_j & \text{for } u_i = 1 \wedge u_j = 1, \end{cases} \tag{13}$$

what can be converted to

$$\begin{aligned} [u_i]P_i + [u_j]P_j = \; & [(1-u_i)(1-u_j)]P + [u_i(1-u_j)]P_i \\ & +[(1-u_i)u_j]P_j + [u_i \cdot u_j](P_i + P_j). \end{aligned} \tag{14}$$

Due to the fact, that in subsequent additions the first argument is a sum of the previous elements and it is different than $\mathcal{O}$, the form of an operation $R_i + P_{i+2}$ is

$$R_i + [u_{i+2}]P_{i+2} = \begin{cases} R_i & \text{for } u_{i+2} = 0, \\ R_i + P_{i+2} & \text{for } u_{i+2} = 1, \end{cases} \tag{15}$$

what is equal to

$$R_i + [u_{i+2}]P_{i+2} = [(1-u_{i+2})]R_i + [u_{i+2}](R_i + P_{i+2}). \tag{16}$$

H. Complexity

As the first idea for generalization requires less operations, the memory complexity will be calculated for that method. The whole process was described in details for Edwards curves in [18, §4.2]. Computation for case with short Weierstrass curves is straightforward to conduct based on the mentioned paper, the differences are in the selected model. The proper addition formulas will be written in Sec. III-A.

Because the computation is technical, tedious and may be conducted based on [18], it will not be described here.

Eventually, the number of binary variables necessary to perform the reduction of ECDLP to the QUBO for short Weierstrass curve is equal to $O\left(\frac{n^3}{\log_2 n}\right)$, where n is the bit length of a field characteristic p and d is the base of a point multiplicity representation, thus the generalization is asymptotically equal to the original method.

III. Practical Example

Because for fields of characteristic greater than 3 every elliptic curve can be transformed into a short Weierstrass curve, this model will be used in the example.

A. Short Weierstrass curve

Short Weierstrass curve E_{SW} over prime field $\mathbb{F}_p$ is given by the equation

$$y^2 = x^3 + ax + b, \tag{17}$$

where a, b are coordinates from $\mathbb{F}_p$.

The addition formulas on short Weierstrass curve E_{SW} for points $P = (x_1, y_1)$, $Q = (x_2, y_2)$ and $R = P + Q = (x_3, y_3)$ are

$$\begin{aligned} x_3 &= \tfrac{(y_2-y_1)^2-(x_1+x_2)(x_2-x_1)^2}{(x_2-x_1)^2} = \tfrac{\text{nom}_x}{\text{denom}_x}, \\ y_3 &= \tfrac{(2x_1+x_2)(y_2-y_1)(x_2-x_1)^2-(y_2-y_1)^3-y_1(x_2-x_1)^3}{(x_2-x_1)^3} \\ &= \tfrac{\text{nom}_y}{\text{denom}_y}. \end{aligned} \tag{18}$$

B. Explanation

The example will be conducted with the first idea from Subsection II-F. To do this, Eq. (12) has to be used. Combining it with addition formulas Eqs. (18), one obtains

$$\begin{aligned} R_i = \; & [u_i]P_i + [u_j]P_j = [u_i(1-u_j)]P_i + [(1-u_i)u_j]P_j \\ & +[u_i \cdot u_j]\tfrac{\text{nom}_{x/y}}{\text{denom}_{x/y}}. \end{aligned} \tag{19}$$

Because in the QUBO model the equation must equal 0 and must not be represented with any fractions, the last transformation is required. After that, the formula is given by

$$\begin{aligned} F_i = \; & u_i \cdot u_j \cdot \text{nom}_{x/y} + \text{denom}_{x/y} \cdot \big(u_i \cdot (1-u_j) \cdot P_{i,x/y} \\ & +u_j \cdot (1-u_i) \cdot P_{j,x/y}\big) - R_{i,x/y} \cdot \text{denom}_{x/y}. \end{aligned} \tag{20}$$

Because the formula in Eq. (20) does not operate on points on an elliptic curve, instead uses polynomials and numerical coordinates, variables u_i and u_j are treated as integers from set $\{0, 1\}$ and thus notation with square brackets is unnecessary and may be confusing.

For every single addition, two functions F_i will be obtained – one for coordinate x and one for coordinate y.

C. Experiment

Consider the following short Weierstrass curve $E_W/\mathbb{F}_3$: $y^2 = x^3+3x+1$. The order of the group of points is equal to 7 and the group is cyclic. The generator of this group is point $P = (2, 1)$ and $Q = (0, 2) = [y]P$.

Since $ord(P) = 7$ and $Q \neq \mathcal{O}$, $y \in \{1, \ldots, 6\}$. The sought multiplicity y can be written as $y = 4u_2 + 2u_1 + u_0$. Based on (3), one get

$$\begin{aligned} [y]P &= [4u_2 + 2u_1 + u_0]P = [4u_2]P + [2u_1]P + [u_0]P \\ &= [u_2]\left([4]P\right) + [u_1]\left([2]P\right) + [u_0]P \\ &= [u_2]P_4 + [u_1]P_2 + [u_0]P_1. \end{aligned} \tag{21}$$

One can compute $P_4 = (1, 1)$, $P_2 = (0, 1)$ and $P_1 = (2, 1)$. Because ECDLP is defined over $\mathbb{F}_3$ and $ord(P) = 7$, only one point R_i will be necessary and it has a form $R_1 = (2u_4 + u_5,\ 2u_6 + u_7)$. From addition $[u_0]P_1 + [u_1]P_2 = R_1$ one obtains

$$\begin{aligned} F_1 = &\quad 2u_0u_1 + 2u_0 + u_4 + 2u_5, \\ F_2 = &\quad u_0 + u_1 + u_6 + 2u_7. \end{aligned}$$

From addition $R_1 + [u_2]P4 = Q$ one obtains

$$\begin{aligned} F_3 = &\quad 2u_2u_4^3 + u_2u_5^3 + 2u_4^3 + u_5^3 + u_2u_6^2 + u_2u_6u_7 + u_2u_7^2 \\ &+ u_4^2 + u_4u_5 + u_5^2 + 2u_2u_6 + u_2u_7 + 2u_4 + u_5, \\ F_4 = &\quad u_2u_4^3 + 2u_2u_5^3 + 2u_4^3u_6 + u_5^3u_6 + 2u_2u_6^3 + u_4^3u_7 \\ &+ 2u_5^3u_7 + u_2u_7^3 + u_4^3 + 2u_5^3 + 2u_6 + u_7 + 1. \end{aligned}$$

Next, the following operations are performed:

1) reducing u^k to u using a property of binary variables,
2) transformation from the pseudo-boolean function over F_3 to the pseudo-boolean function over integers,
3) linearization,
4) summing squares of all equations.
5) adding penalties.

More details about the whole process can be found in [8].

The correct solution was found, which is $y = 5$ $(u_2, u_1, u_0) = (1, 0, 1)$. The values of parameters used in solving this QUBO problem are shown in Tab. III-C. Connections between source variables are presented in Fig. 3. The embedding of a problem to D-Wave Advantage is presented in Fig. 4.

TABLE I
D-WAVE ADVANTAGE SOLVER PARAMETERS USED IN SOLVING QUBO PROBLEM EQUIVALENT TO THE PROBLEM OF SOLVING ECDLP OVER $\mathbb{F}_3$ ON SHORT WEIERSTRASS CURVE IN A SUBGROUP OF SIZE 7

Parameter	Value
Solver	Advantage2_prototype2.3
Qubits	1248
Topology	Zephyr
Number of read	10000
Annealing Time	200 µs
Number of source variables	25
Number of target variables	51
Max chain length	3

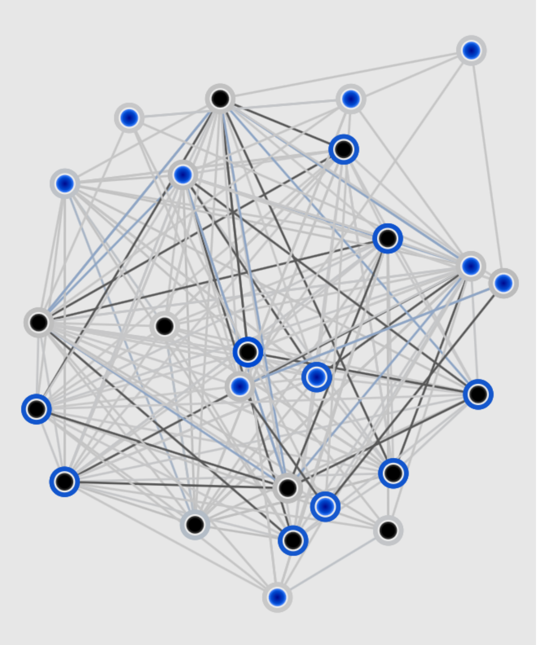

Fig. 3. Connection between source variables

IV. Conclusion and Further Work

This paper presents a generalization of a method for solving the elliptic curve discrete logarithm problem with quantum annealing, described in [17]. The proper work of the method was confirmed with the practical implementation on the D-Wave computer with the D-Wave Leap cloud [19].

The ECDLP has been solved with a fully quantum device for a 2-bit finite field. Unfortunately, bigger problems have not been solved fully quantum. However, a few more problems have been solved with a hybrid solver. The biggest solved problem was ECDLP on an elliptic curve with order 7 over a 4-bit field $\mathbb{F}_{11}$. Detailed information may be found in the below Tab. IV. About the colors of the cells in a table:

- the green ones mean that the problem has been solved with a fully-quantum solver,
- the yellow ones mean that the problem has been solved with a hybrid solver,
- the white one that problem has not been solved.

TABLE II
NUMBER OF BINARY VARIABLES NECESSARY TO SOLVE ECDLP ON ELLIPTIC CURVE WITH ORDER 7

Finite field bit length [b]	Finite field	1^{st} method	2^{nd} method
2	$\mathbb{F}_3$	26	32
3	$\mathbb{F}_5$	74	98
3	$\mathbb{F}_7$	74	98
4	$\mathbb{F}_{11}$	159	202

The advantages of the generalized method:

- does not require the numerical form of the neutral element,
- does not require using projective coordinates,
- allows to solve ECDLP on any model of elliptic curve,

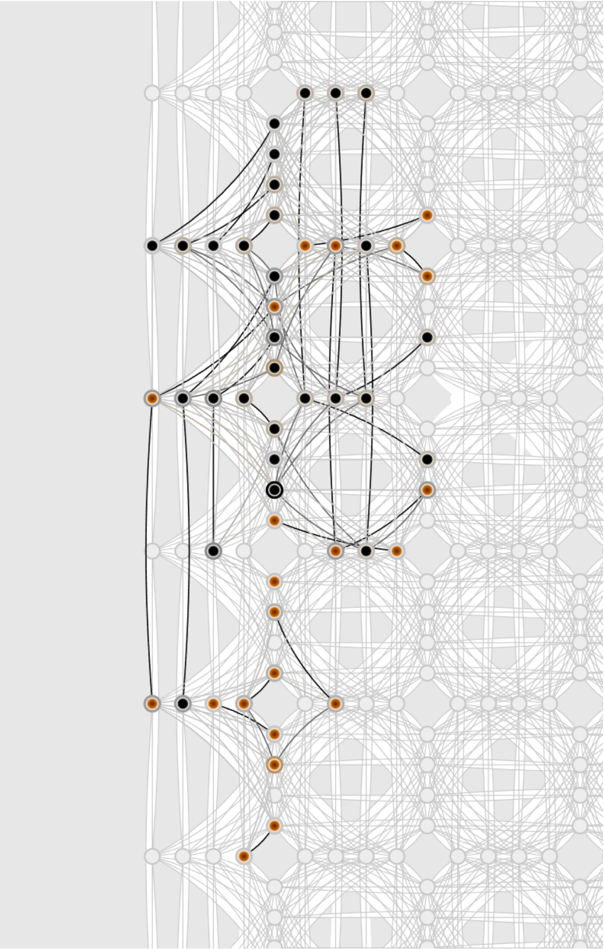

Fig. 4. Embedding of a problem equivalent to the problem of finding elliptic curve discrete logarithm over $\mathbb{F}_3$ on shorts Weierstrass curve to the D-Wave Advantage

- proper working is confirmed with quantum and hybrid computations, as well as with simulated annealing.

The disadvantages of the new method:

- requires more resources than the original method, but is asymptotically equal to it.

As seen in Tab. IV, the first method requires fewer variables to solve ECDLP than the second one, however it demands supplementary actions for multiplicities divisible by 4. The alternative method does not need additional effort, but this comes at the cost of more necessary resources. Considering those arguments and the fact, that the extraordinary cases in a simpler method will happen with an expected probability about 25%, the author recommends using the first method.

The description of the generalized method does not exhaust a topic. In further work, some improvements may be tried, for example:

- changing the multiplicity representation like in [18],
- checking described method for other elliptic curve models,
- checking if for special cases using morphisms and arithmetic on other models will allow to use less resources like in [20].

References

[1] S. Pohlig and M. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.)," *IEEE Transactions on information Theory*, vol. 24, no. 1, pp. 106–110, 1978.

[2] D. Shanks, "Five number-theoretic algorithms," in *Proceedings of the Second Manitoba Conference on Numerical Mathematics (Winnipeg), 1973*, 1973.

[3] D. Bernstein and T. Lange, "Two grumpy giants and a baby," *The Open Book Series*, vol. 1, no. 1, pp. 87–111, 2013.

[4] J. M. Pollard, "Monte Carlo methods for index computation (mod p)," *Mathematics of computation*, vol. 32, no. 143, pp. 918–924, 1978.

[5] L. C. Washington, *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.

[6] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum resource estimates for computing elliptic curve discrete logarithms," 2017. [Online]. Available: https://arxiv.org/abs/1706.06752

[7] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[8] M. Wroński, "Practical solving of discrete logarithm problem over prime fields using quantum annealing," in *International Conference on Computational Science*. Springer, 2022, pp. 93–106.

[9] 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.

[10] E. Burek and M. Wroński, "Quantum Annealing and Algebraic Attack on Speck Cipher," in *International Conference on Computational Science*. Springer, 2022, pp. 143–149.

[11] M. Wroński, E. Burek, and M. Leśniak, "(In) security of stream ciphers against quantum annealing attacks on the example of the Grain 128 and Grain 128a ciphers," *Cryptology ePrint Archive*, 2023.

[12] 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, 2018.

[13] W. Peng, B. Wang, F. Hu, Y. Wang, X. Fang, X. Chen, and C. Wang, "Factoring larger integers with fewer qubits via quantum annealing with optimized parameters," *SCIENCE CHINA Physics, Mechanics & Astronomy*, vol. 62, pp. 1–8, 2019.

[14] O. Żołnierczyk and M. Wroński, "Searching B-Smooth Numbers Using Quantum Annealing: Applications to Factorization and Discrete Logarithm Problem," in *International Conference on Computational Science*. Springer, 2023, pp. 3–17.

[15] J. Ding, G. Spallitta, and R. Sebastiani, "Effective prime factorization via quantum annealing by modular locally-structured embedding," *Scientific Reports*, vol. 14, no. 1, p. 3518, 2024.

[16] M. Wroński, "Index calculus method for solving elliptic curve discrete logarithm problem using quantum annealing," in *International Conference on Computational Science*. Springer, 2021, pp. 149–155.

[17] 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.

[18] 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.

[19] D-Wave Advantage Leap. [Online]. Available: https://cloud.dwavesys.com/leap/

[20] M. Wroński, "Faster point scalar multiplication on short Weierstrass elliptic curves over Fp using twisted Hessian curves over Fp2," *Journal of Telecommunications and Information Technology*, no. 3, pp. 98–102, 2016.