Transformation of Elliptic Curve Discrete Logarithm Problem to QUBO Using Direct Method in Quantum Annealing Applications

Michał Wroński¹, Elżbieta Burek², Łukasz Dzierzkowski², and Olgierd Żołnierczyk²

¹Department of Cryptology, NASK National Research Institute, Warsaw, Poland,
²Faculty of Cybernetics, Military University of Technology, Warsaw, Poland

https://doi.org/10.26636/jtit.2024.1.1463

Abstract — This paper investigates how to reduce the elliptic curve discrete logarithm problem over prime fields to the quadratic unconstrained binary optimization (QUBO) problem in order to obtain as few logical qubits as possible. In the best case scenario, if \( n \) is the bitlength of a characteristic of prime field \( \mathbb{F}_p \), approximately \( 3n^3 \) logical qubits are required for such a reduction in the Edwards curve case. We present a practical attack on an elliptic curve discrete logarithm problem over the 3-bit prime field \( \mathbb{F}_2 \); for an elliptic curve with the subgroup of order 8. We solved this problem using the D-Wave Advantage QPU. To the best of the authors’ knowledge, no one has made, so far, a practical attack on the elliptic curve discrete logarithm over a prime field using the direct quantum method.

Keywords — cryptanalysis, D-Wave, elliptic curve discrete logarithm problem, quantum annealing

1. Introduction

Shor’s quantum algorithm for factorization and discrete logarithm computation [1] is one of the essential research areas in modern cryptology. The resources required to run Shor’s quantum algorithm have been widely analyzed in [2]–[4]. The estimations of resources required for implementing Shor’s algorithm for the elliptic curve discrete logarithm problem (ECDLP) are presented in [5] for binary elliptic curves and in [6] for elliptic curves over prime fields.

As one may notice from Tab. 1, for real-world security parameters, the number of qubits necessary to run Shor’s algorithm is not so big, but the number of Toffoli gates is huge. From this point of view, much work is still required to run Shor’s algorithm to solve ECDLP for real-world security parameters. It is worth noting that a recent review of ECDLP in modern cryptology. The resources required to run Shor’s algorithm computation [1] is one of the essential research areas in modern cryptology. The resources required to run Shor’s quantum algorithm have been widely analyzed in [2]–[4]. The estimations of resources required for implementing Shor’s algorithm for the elliptic curve discrete logarithm problem (ECDLP) are presented in [5] for binary elliptic curves and in [6] for elliptic curves over prime fields.

As one may notice from Tab. 1, for real-world security parameters, the number of qubits necessary to run Shor’s algorithm is not so big, but the number of Toffoli gates is huge. From this point of view, much work is still required to run Shor’s algorithm to solve ECDLP for real-world security parameters. It is worth noting that a recent review of ECDLP [5] for binary elliptic curves and in [6] for elliptic curves over prime fields.

The QUBO problem may also be viewed as a problem of minimizing a function such as:

\[
\min_{x \in \{0,1\}^N} x^T Q x, \quad (1)
\]

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

The QUBO problem may also be viewed as a problem of minimizing a function such as:

\[
f(x) = \sum_i Q_i x_i + \sum_{i<j} Q_{i,j} x_i x_j. \quad (2)
\]

Let us note that the QUBO problem is a special case of the binary quadratic model (BQM) problem, where BQM may be given as:

\[
\sum_i a_i v_i + \sum_{i<j} b_{i,j} v_i v_j + c, \quad (3)
\]

with \( a_i \) and \( b_{i,j} \) being real numbers and \( v_i \in \{-1,+1\} \) or \( \{0,1\} \). Transformation of the QUBO problem to the BQM problem for \( v_i \in \{0,1\} \) is straightforward – we must forget the constant \( c \) appearing in BQM.

This paper shows how to transform the ECDLP over prime fields to the QUBO problem. The best method allows to convert a discrete logarithm problem over a prime field \( \mathbb{F}_p \) to the QUBO problem using approximately \( 3n^3 \) logical qubits, where \( n \) is the bitlength of \( p \).

With the scope of researched defined above, the authors’ contribution consists in the following:

- presenting a method for reducing the elliptic curve discrete logarithm problem to the QUBO problem, with the said method requiring approximately \( 3n^3 \) logical qubits for such a reduction,

   fields [9] into the quadratic unconstrained binary optimization (QUBO) problem, and then solving this problem using the D-Wave computer.

QUBO [10] is a significant problem with many real-world applications. The QUBO model can be described by the following optimization problem:

\[
\min_{x \in \{0,1\}^N} x^T Q x, \quad (1)
\]

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

The QUBO problem may also be viewed as a problem of minimizing a function such as:

\[
f(x) = \sum_i Q_i x_i + \sum_{i<j} Q_{i,j} x_i x_j. \quad (2)
\]

Let us note that the QUBO problem is a special case of the binary quadratic model (BQM) problem, where BQM may be given as:

\[
\sum_i a_i v_i + \sum_{i<j} b_{i,j} v_i v_j + c, \quad (3)
\]

with \( a_i \) and \( b_{i,j} \) being real numbers and \( v_i \in \{-1,+1\} \) or \( \{0,1\} \). Transformation of the QUBO problem to the BQM problem for \( v_i \in \{0,1\} \) is straightforward – we must forget the constant \( c \) appearing in BQM.

This paper shows how to transform the ECDLP over prime fields to the QUBO problem. The best method allows to convert a discrete logarithm problem over a prime field \( \mathbb{F}_p \) to the QUBO problem using approximately \( 3n^3 \) logical qubits, where \( n \) is the bitlength of \( p \).

With the scope of researched defined above, the authors’ contribution consists in the following:

- presenting a method for reducing the elliptic curve discrete logarithm problem to the QUBO problem, with the said method requiring approximately \( 3n^3 \) logical qubits for such a reduction,
• presenting a practical example and the results of solving ECDLP on the Edwards curve over $\mathbb{F}_7$ for the problem with the order of generator equal to 8, using the D-Wave Advantage QPU.

It is worth noting that, to the best of the authors’ knowledge, no one has ever made a practical attack on the elliptic curve discrete logarithm over a prime field using direct quantum methods. Relying on the index calculus method, the application of quantum annealing to solving ECDLP over prime fields was presented in [11]. In that paper, hybrid classical-quantum annealing methods were used to collect relations. After that, the linear algebra step was computed classically to retrieve the private key.

The results presented in this paper are more connected with [9], where the DLP problem was transformed directly to the QUBO problem and then solved using the direct quantum annealing method. The main results of this paper are showing the direct transformation of the ECDLP on Edwards curves to the QUBO problem, with this problem then being solved using direct quantum annealing methods.

Solving such a QUBO problem is equivalent to retrieving the private key. There is no need to solve linear algebra steps, as was required in [11]. However, it is also worth noting that any QUBO problem may be run on a computer using a classical annealing algorithm (for example simulated annealing). Yet, there are some heuristic arguments that, in many cases, the complexity of quantum annealing may reach even $e^{\sqrt{N}}$ for a QUBO problem consisting of $N$ variables [12], with the simulated annealing algorithm offering some gains, as its complexity is $e^N$. Therefore, quantum annealing is much more interesting to consider in this case. However, the presented example is of the minor variety and the results obtained constitute another step in applying quantum computations to classical public-key cryptography problems.

2. Transformation of General Discrete Logarithm Problem to QUBO Problem

Consider a general discrete logarithm problem in any cyclic finite group $G$ with group operation $\circ$. Let us also denote scalar multiplication by $y$ in $G$ by $y \circ g$ as:

$$y : G \rightarrow G,$$

$$y \circ g = g \circ \cdots \circ g,$$  \hspace{1cm} (4)

Now, the discrete logarithm problem is defined. Having elements $g, h \in G$ for which holds that:

$$h = y \circ g,$$ \hspace{1cm} (5)

the problem is to find proper $y$.

If for any $g_1, g_2 \in G$ holds that $g_1 \circ g_2$ may be written as a multivariate Boolean polynomial with integer coefficients, then such a discrete logarithm problem may be transformed to the QUBO problem.

Let $n$ be the bitlength of the order of element $g$, denoted as $\text{ord}(g)$.

If $y = 2^{m-1}u_m + \cdots + 2u_2 + u_1$, where $u_1, \ldots, u_m$ are binary variables. Then:

$$y \circ g = (2^{m-1}u_m + \cdots + 2u_2 + u_1) \circ g$$

$$= (2^{m-1}u_m) \circ \cdots \circ (2u_2) \circ (u_1) \circ g$$

$$= (u_m \circ (2^{m-1} \circ g)) \circ \cdots \circ (u_2 \circ (2 \circ g)) \circ (u_1 \circ g).$$ \hspace{1cm} (6)

Let $a$ be the neutral element of group operation in $G$. Then, for every element $g \in G$ must hold:

$$u_i \circ (2^{i-1} \circ g) = \begin{cases} a, & u_i = 0, \\ 2^{i-1} \circ g, & u_i = 1. \end{cases}$$ \hspace{1cm} (7)

The most important factor in this context is to ensure that one will be able to write $u_i \circ (2^{i-1} \circ g)$ as in Eq. (7), using multivariate polynomials of Boolean variables and real coefficients. By applying $x_i = u_i \circ (2^{i-1} \circ g)$, the general DLP problem given by Eq. (5) may be transformed to a problem of finding the solution of:

$$x_1 \circ x_2 \circ \cdots \circ x_m = h.$$ \hspace{1cm} (8)

Transformation of the discrete logarithm problem over finite fields in additive and multiplicative groups has been presented in detail in [9]. Application of the quantum annealing approach while solving such discrete logarithms was also presented. Therefore, we omit these descriptions and focus mainly on transforming the elliptic curve discrete logarithm problem to the QUBO problem and solving it using quantum annealing.

3. Transformation of Elliptic Curve Discrete Logarithm Problem to QUBO

Elliptic curve cryptography (ECC) is an important part of modern security. Many cryptographic problems are based on the computational complexity of the elliptic curve discrete logarithm problem (ECDLP), which will be described below. The most popular cryptographic algorithms based on ECDLP are Elliptic Curve Diffie-Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA).

<p>| Tab. 1. Estimated number of qubits and Toffoli gates for Shor’s attack on ECDLP. |
|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>Bitlength of the base field</th>
<th>Number of qubits</th>
<th>Number of Toffoli gates</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1176</td>
<td>$1.52 \cdot 10^{10}$</td>
</tr>
<tr>
<td>192</td>
<td>1754</td>
<td>$5.30 \cdot 10^{10}$</td>
</tr>
<tr>
<td>256</td>
<td>2330</td>
<td>$1.29 \cdot 10^{11}$</td>
</tr>
<tr>
<td>384</td>
<td>3484</td>
<td>$4.49 \cdot 10^{11}$</td>
</tr>
<tr>
<td>512</td>
<td>4636</td>
<td>$1.09 \cdot 10^{12}$</td>
</tr>
</tbody>
</table>
We begin by defining an elliptic curve discrete logarithm problem over a prime field $\mathbb{F}_p$:
\[ [y]P = Q , \]  
where $P, Q \in E(\mathbb{F}_p)$ and $y \in \{1, \text{ord}(P) - 1\}$.

Let $m$ be the bitlength of $\text{ord}(P)$.

If $y = 2^{m-1}u_m + \cdots + 2u_2 + u_1$, where $u_1, \ldots, u_m$ are binary variables. Then:
\[ [y]P = [2^{m-1}u_m + \cdots + 2u_2 + u_1]P = [2^{m-1}u_m]P + \cdots + [2u_2]P + [u_1]P = [u_m][2^{m-1}]P + \cdots + [u_2][2]P + [u_1]P . \]

It is worth noting that writing
\[ y = 2^{m-1}u_m + \cdots + 2u_2 + u_1 \]
allows to obtain $y > \text{ord}(P)$, and to get the result from $\{0, \ldots, \text{ord}(P) - 1\}$ computing $y \mod \text{ord}(P)$.

For simplicity, let us suppose that the neutral element $\mathcal{O}$ of addition operation in $E(\mathbb{F}_p)$ may be represented using affine coordinates and $\mathcal{O} = (O_x, O_y)$. This is the case, for instance, in Edwards or twisted Edwards curves. Then, for every point $P = (P_x, P_y) \in E(\mathbb{F}_p)$:
\[ [u_i][2^{i-1}]P = \begin{cases} \mathcal{O}, & u_i = 0 , \\ [2^{i-1}]P, & u_i = 1 . \end{cases} \]

If $P_i = [u_i][2^{i-1}]P$, Eq. (11) above is equivalent to:
\[ \begin{align*} P_{i,x} &= O_x + u_i \left( [2^{i-1}]P_x - O_x \right) , \\ P_{i,y} &= O_y + u_i \left( [2^{i-1}]P_y - O_y \right) . \end{align*} \]

Similar transformations can be repeated when the neutral element cannot be represented using affine coordinates but must be presented using projective coordinates, as it is the case in short Weierstrass curves.

Now, the ECDLP given by Eq. (9) may be transformed to the problem of finding a solution of:
\begin{align*}
& \begin{align*}
& P_1 + P_2 + \cdots + P_m = Q . \tag{13}
& \end{align*}
& \end{align*}

Let $E$ be an elliptic curve with complete arithmetic and let us assume, for simplicity, that all points from $\{P\}$ may be presented using affine coordinates. Then, for every two points $P_1, P_2 \in E(\mathbb{F}_p)$ and point $Q \in E(\mathbb{F}_p)$, where $Q = P_1 + P_2$ holds:
\[ \begin{align*}
& \begin{align*}
& Q_x = \phi(P_1, P_2) , \\
& Q_y = \psi(P_1, P_2) , \tag{14}
& \end{align*}
& \end{align*}\]

where $\phi, \xi, \psi$ are polynomials.

To solve the problem given by Eq. (13), a regular binary tree of maximal height is used (Fig. 2), the same as for transforming the DLP problem to the QUBO problem [9].

In Eq. (14), point $Q$ is computed explicitly, but may be presented by the coordinates of every point implicitly.

For the case of ECDLP and the method of decomposition presented in Fig. 2, the following system of equations is obtained:
\[ \begin{align*}
& \begin{align*}
& f_{1,1} = \phi(P_1, P_2) - R_{1,x} \psi(P_1, P_2) \mod p - k_{1,1}p = 0 , \\
& f_{1,2} = \xi(P_1, P_2) - R_{1,y} \psi(P_1, P_2) \mod p - k_{1,2}p = 0 , \\
& f_{2,1} = \phi(R_1, P_3) - R_{2,x} \psi(R_1, P_3) \mod p - k_{2,1}p = 0 , \\
& f_{2,2} = \xi(R_1, P_3) - R_{2,y} \psi(R_1, P_3) \mod p - k_{2,2}p = 0 , \\
& \cdots \\
& f_{m-2,1} = \phi(R_{m-3}, P_{m-1}) - R_{m-2,x} \psi(R_{m-3}, P_{m-1}) \mod p - k_{m-2,1}p = 0 , \\
& f_{m-2,2} = \xi(R_{m-3}, P_{m-1}) - R_{m-2,y} \psi(R_{m-3}, P_{m-1}) \mod p - k_{m-2,2}p = 0 , \\
& f_{m-1,1} = \phi(R_{m-3}, P_{m-1}) - Q_x \mod p - k_{m-1,1}p = 0 , \\
& f_{m-1,2} = \xi(R_{m-3}, P_{m-1}) - Q_y \mod p - k_{m-1,2}p = 0 , \tag{15}
& \end{align*}
& \end{align*}\]
where:

\[ P_i = (P_{ix}, P_{iy}) \text{ for } i = 1, m \] and

\[ R_i = (R_{ix}, R_{iy}) \text{ for } i = 1, m - 2. \]

Next, we will focus on the transformation of the ECDLP defined on different models of elliptic curves to the QUBO problem.

In the direct method presented above, it is important to use only one kind of formula that should work for all proper possible inputs. In the case of the elliptic curve in the Weierstrass form, the main issue is that efficient formulas work separately while adding and separately while doubling. Furthermore, in the case of the Weierstrass curve, a neutral element cannot be added using classical, reasonably efficient addition formulas. On the other hand, complete arithmetic formulas exist for the Weierstrass curves, but they are inefficient [13]. Therefore, using an elliptic curve model that would allow to use the following:

- efficient arithmetic with a small number of multiplications,
- neutral element which can be represented by affine coordinates,
- complete arithmetic, seems much more convenient.

Therefore, we focus on applying the proposed method to the case of Edwards curves, which fulfills all the conditions presented above.

3.1. Edwards Curves

**Definition 1.** The Edward's curve \( E_{Ed} \) over a field \( \mathbb{K} \) is given by [14]:

\[ E_{Ed}/\mathbb{K} : x^2 + y^2 = 1 + dx^2y^2, \] (16)

where \( d \not\in \{0, 1\} \).

The sum of points \( P = (x_1, y_1) \) and \( Q = (x_2, y_2) \) on \( E_{Ed} \) is given by:

\[ P + Q = \left( \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right). \] (17)

The neutral element is \( O = (0, 1) \) and the negation is given by \(- (x, y) = (-x, y)\). If \( d \) is not a square in \( \mathbb{K} \), the above addition formula is complete in the \( \mathbb{K} \)-rational points on \( E \).

Putting \( P_i = [u_i]([2^{i-1}]P) \), the Eq. (12) for Edwards curves is equivalent to:

\[ \begin{align*}
    P_{i,x} &= u_i \left(\left([2^{i-1}]P\right)_x, \\
    P_{i,y} &= 1 + u_i \left(\left([2^{i-1}]P\right)_y - 1\right). 
\end{align*} \] (18)

In such a case, using the idea presented earlier as well as Eq. (15), the system of equations in the case of the Edwards curve will be given by Eq. (19). Assume that the Edwards curve is defined over a prime field \( \mathbb{F}_p \) and \( n \) denotes the bitlength of \( p \) and \( m \) denotes the bitlength of the size of the order of the group generated by \( P \).

\[ f_{i,1} = (A_1 - P_{1,y}P_{2,y}) \mod p - k_{1,p} = 0, \]

\[ f_{i,2} = (B_1 - P_{1,x}P_{2,x}) \mod p - k_{1,2p} = 0, \]

\[ f_{i,3} = (C_1 - P_{1,x}P_{2,y}) \mod p - k_{1,3p} = 0, \]

\[ f_{i,4} = (D_1 - P_{1,y}P_{2,x}) \mod p - k_{1,4p} = 0, \]

\[ f_{i,5} = (E_1 - C_1D_1) \mod p - k_{1,5p} = 0, \]

\[ f_{i,6} = (F_1 - R_{1,x}E_1) \mod p - k_{1,6p} = 0, \]

\[ f_{i,7} = (G_1 - R_{1,y}E_1) \mod p - k_{1,7p} = 0, \]

\[ f_{i,8} = (C_1 + D_1 - R_{1,1, y} - P_1) \mod p - k_{1,8p} = 0, \]

\[ f_{i,9} = (A_1 + B_1 + Q_{1,y} - G_1) \mod p - k_{1,9p} = 0, \]

\[ f_{m-1,1} = (A_{m-1} - P_{m, y}R_{m-2,y}) \mod p - k_{m-1,1} = 0, \]

\[ f_{m-2,2} = (B_{m-2} - P_{m,x}R_{m-2,x}) \mod p - k_{m-2,2p} = 0, \]

\[ f_{m-3,3} = (C_{m-2} - P_{m,x}R_{m-2,y}) \mod p - k_{m-3,3p} = 0, \]

\[ f_{m-4,4} = (D_{m-2} - P_{m,y}R_{m-2,x}) \mod p - k_{m-4,4p} = 0, \]

\[ f_{m-5,5} = (E_{m-2} - C_{m-2}D_{m-2}) \mod p - k_{m-5,5p} = 0, \]

\[ f_{m-6,6} = (F_{m-2} - R_{m,x}E_{m-2}) \mod p - k_{m-6,6p} = 0, \]

\[ f_{m-7,7} = (G_{m-2} - Q_{m,y}E_{m-2}) \mod p - k_{m-7,7p} = 0, \]

\[ f_{m-8,8} = (C_{m-2} + D_{m-2} - Q_{x} - F_{m-2}) \mod p - k_{m-8,8p} = 0. \] (19)

Let us compute the number of logical variables required to transform the ECDLP problem on the Edwards curve into the QUBO problem. We begin by counting the necessary variables for a single system of equations \( f_{1,1}, \ldots, f_{1,9} \), where \( 1 < i < m - 2 \). Due to the fact that cases for \( i = 1 \) and \( i = m - 2 \) are similar, the analysis for these two scenarios may be omitted.

Consider \( f_{i,1} \) case in the number of variable context. For variables \( A_i \), \( n \) bits are necessary to represent them because every \( A_i \) is in the set \( \{1, \ldots, p - 1\} \). During the multiplication of \( P_{i+1,y} \) and \( R_{i-1,y} \), there will be \( n \) monomials of degree \( 2 \) (\( P_{i-1,y} \) consists of two terms, but only one Boolean variable) and \( n \) monomials of degree \( 1 \). This means that \(-P_{i+1,y}R_{i-1,y} \mod p \) will consist, in such a case, of \( 2n \) monomials with coefficients from set \( \{0, \ldots, p - 1\} \). Finally, it means that the maximum value of polynomial \( f_{i,1} \) is equal to \( (2n + 1)(p - 1) \), which means that \( k_{i,1} \leq (2n + 1)(p - 1) < 2n + 1 \), then \( k_{i,1} < 2n \) and the bilength of \( k_{i,1} \) is equal to \( \lfloor \log_2(2n) \rfloor + 1 \) at most. Hence, for equation \( f_{i,1} \), we have:

- \( n \) additional variables obtained during linearization of square monomials,
The case concerned with equation $f$ with coefficients from set $\{0, \ldots, p-1\}$. Therefore, for equation $f_{1,1}$, $2n + \lfloor \log_2 (2n) \rfloor + 1$ additional variables are necessary. The same applies to equation $f_{1,4}$.

Let us focus on equation $f_{1,2}$. For variables $B_i$, $n$ bits are required to represent them, because every $B_i$ is in set $\{0, \ldots, p-1\}$. On the other hand, during the multiplication of terms $P_{i+1, x}R_{i-1, x}$ there will be $n$ monomials of degree 2, because $R_{i-1, x}$ consists of one term and only one Boolean variable. This means that $(-P_{i+1, x}R_{i-1, x}) \mod p$ will consist, in such a case, of $n$ monomials with coefficients from set $\{0, \ldots, p-1\}$. It means that the maximum value of polynomial $f_{1,2}$ is equal to $(n+1)(p-1)$, because the value of $B_i$ is also limited by $p-1$.

Therefore, $k_{i,2}p \leq (n+1)(p-1)$, which means that $k_{i,2} \leq \frac{(n+1)(p-1)}{p} < n + 1$, so $k_{i,2} \leq n$ and the bitlength of $k_{i,2}$ is equal to $\lfloor \log_2 (n) \rfloor + 1$ at most. So, for equation $f_{1,2}$, we have:

- $n$ additional variables obtained during linearization of square monomials,
- $n$ Boolean variables for variable $B_i$,
- $\lfloor \log_2 (2n) \rfloor + 1$ Boolean variables necessary for writing variable $k_{i,1}$.

Therefore, for equation $f_{1,2}$, $2n + \lfloor \log_2 (n) \rfloor + 1$ additional variables are necessary. The same results apply to equation $f_{1,4}$.

As far as equation $f_{1,5}$ is concerned, $n$ bits are necessary to represent variables $E_i$, because every $E_i$ is in set $\{1, \ldots, p-1\}$. On the other hand, during the multiplication of terms $C_i$ and $D_i$, there will be $n^2$ monomials of degree 2 (both $C_i$, $D_i$ consist of $n$ Boolean variables). This means that $(-C_iD_i) \mod p$ will consist, in such a case, of $n^2$ monomials with coefficients from set $\{0, \ldots, p-1\}$. Finally, it means that the maximum value of polynomial $f_{1,5}$ is $(n^2+1)(p-1)$, because the value of $E_i$ is also limited by $p-1$.

Therefore, $k_{i,5}p \leq (n^2+1)(p-1)$, which means that $k_{i,5} \leq \frac{(n^2+1)(p-1)}{p} < n^2 + 1$, so $k_{i,5} \leq n^2$ and the bitlength of $k_{i,5}$ is equal to $\lfloor \log_2 (n^2) \rfloor + 1$ at most. So, for equation $f_{1,5}$, we have:

- $n^2$ additional variables obtained during linearization of square monomials,
- $n$ Boolean variables for variable $E_i$,
- $\lfloor \log_2 (n^2) \rfloor + 1$ Boolean variables necessary for writing variable $k_{i,5}$.

This proves that, for equation $k_{i,5}$, $n^2 + n + \lfloor \log_2 (n^2) \rfloor + 1$ additional variables are necessary.

The same applies to equations $f_{1,6}$ and $f_{1,7}$, but in each of these cases $n$ bits are necessary for representation of $R_{i,x}$ and $R_{i,y}$, so for $f_{1,6}$ and $f_{1,7}$, $n^2 + 2n + \lfloor \log_2 (n^2) \rfloor + 1$ additional variables are necessary for each equation.

The case concerned with equation $f_{1,8}$ is the simplest. There are no necessary additional Boolean variables for linearizing square monomials and new variables from a finite field. The only additional variables are necessary for $k_{i,8}$. Let us note that $(-R_{i,x} - F_i) \mod p$ will consist, in such a case, of $2n$ monomials with coefficients from set $\{0, \ldots, p-1\}$. Finally, it means that the maximum value of polynomial $f_{1,8}$ is $(2n+2)(p-1)$, because the values of $C_i$ and $D_i$ are also limited by $p-1$.

Therefore, $k_{i,8}p \leq (2n+2)(p-1)$, which means that $k_{i,8} \leq \frac{(2n+2)(p-1)}{p} < 2n + 2$, so $k_{i,8} < 2n + 1$ and the bitlength of $k_{i,8}$ is equal to $\lfloor \log_2 (2n+1) \rfloor + 1$ at most.

Similar considerations apply to equation $f_{1,9}$. There are no necessary additional Boolean variables for linearizing square monomials and new variables from a finite field. The only additional variables are necessary for $f_{1,9}$. Let us note that $(-G_i) \mod p$ will consist, in such a case, of $n$ monomials with coefficients from set $\{0, \ldots, p-1\}$. Finally, it means that the maximum value of polynomial $f_{1,9}$ is equal to $(n+3)(p-1)$, because the values of $A_i$, $B_i$ and $R_{i,y}$ are also limited by $p-1$.

Hence, $k_{i,9}p \leq (n+3)(p-1)$, because $k_{i,9} \leq \frac{(n+3)(p-1)}{p} < n + 3$, so $k_{i,9} \leq n + 2$ and the bitlength of $k_{i,9}$ is $\lfloor \log_2 (n+2) \rfloor + 1$ at most.

Summing up, for the system of equations $f_{1,1}, \ldots, f_{1,9}$ there are:

$$2 \cdot (2n + \lfloor \log_2 (2n) \rfloor + 1) + 2 \cdot (2n + \lfloor \log_2 (n) \rfloor + 1) + \lfloor \log_2 (n^2) \rfloor + 1 + \lfloor \log_2 (n+2) \rfloor + 1 = 3n^2 + 13n + O(\log_2 n)$$

necessary logical variables.

In the case of the $f_{1,1}, \ldots, f_{1,9}$ system of equations and the $f_{m-2,1}, \ldots, f_{m-2,9}$ system, the number of necessary binary variables is lower. However, it does not influence the overall number of necessary variables much, so we use the same estimations as for $1 < i < m-2$.

Finally, we can estimate the overall number of $i$ Boolean variables for the $f_{1,1}, \ldots, f_{m-2,9}$ as $(m-2) \cdot (3n^2 + 13n + O(\log_2 n))$. If $m \approx n$, which often holds in cryptographic applications, it is approximately $3n^3$.

### 4. Practical Example and Results

Consider the following Edwards curve $E_{Ed}/\F_7 : x^2 + y^2 = 1 + 6x^2y^2$. The order of the group of points of this curve is equal to 8, and the group is cyclic. The generator of this group is point $P = (3, 3)$ and ECDLP with $Q = (4, 3) = [y]P$. We aim to break this ECDLP by finding proper $y$. First, we show how to transform this problem into the QUBO problem.

In the case of small problems, the transformation of entire equations may be more efficient than a separate transformation of each multiplication. Since we know that the order of $P$ is 8 and $Q$ is not the neutral point, $y \in \{1, \ldots, 7\}$. Using binary variables $u_1, u_2, u_3$, we can write $y$ as $y = u_1 + 2u_2 + 4u_3$. 

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 1/2024

79
and, hence:
\[
\]

with \( P_1 = P, P_2 = [2]P \) and \( P_3 = [4]P \), we obtain:
\[
[u_1]P_1 + [u_2]P_2 + [u_3]P_3 = Q. \tag{21}
\]

Let us note that according to Eq. (18) and knowing that \( P_1 = (3, 3), P_2 = (1, 0) \) and \( P_3 = (0, 6) \), we can write
\[
(x_1, y_1) = [u_1]P_1, (x_2, y_2) = [u_2]P_2, (x_3, y_3) = [u_1]P_1 + [u_2]P_2, (x_4, y_4) = [u_3]P_3, (x_5, y_5) = Q \text{ and then:}
\]
\[
\begin{align*}
x_1 &= u_1 (P_{1,x}) = 3u_1, \\
y_1 &= 1 + u_1 (P_{1,y} - 1) = 2u_1 + 1, \\
x_2 &= u_2 (P_{2,x}) = u_2, \\
y_2 &= 1 + u_2 (P_{2,y} - 1) = 6u_2 + 1, \\
x_3 &= u_3 + 2u_4 + 4u_5, \\
y_3 &= u_6 + 2u_7 + 4u_8, \\
x_4 &= u_3 (P_{3,x}) = 0, \\
y_4 &= 1 + u_3 (P_{3,y} - 1) = 5u_9 + 1, \\
x_5 &= 4, \\
y_5 &= 3.
\end{align*}
\tag{22}
\]

Using equations for point addition, because \((x_3, y_3) = (x_1, y_1) + (x_2, y_2)\), the following equations hold:
\[
\begin{align*}
F_1 &= (1 + dx_1x_2y_1y_2)x_3 + 6(x_1y_2 + y_1x_2) = 0, \\
F_2 &= (1 - dx_1x_2y_1y_2)y_3 + 6(y_1y_2 - x_1x_2) = 0.
\end{align*}
\tag{23}
\]

Similarly, because \((x_5, y_5) = (x_3, y_3) + (x_4, y_4)\), the following equations hold:
\[
\begin{align*}
F_3 &= (1 + dx_3x_4y_3y_4)x_5 + 6(x_3y_4 + y_3x_4) = 0, \\
F_4 &= (1 - dx_3x_4y_3y_4)y_5 + 6(y_3y_4 - x_3x_4) = 0.
\end{align*}
\tag{24}
\]

The equations above will then be equal to:
\[
\begin{align*}
F_1 &= 6u_1^2 + 9u_1^2u_3 + 5u_1^2u_2u_4 + 3u_1^2u_2^2u_5 + u_1^2u_2u_3u_8 + 2u_2u_4u_5 + 4u_1u_2u_5 + 3u_1u_2^2u_5 + 6u_1u_2u_4 + 5u_1u_2u_5 + 4u_1u_2u_3 + u_1u_2u_4 + 2u_1u_2u_5 + u_1u_2u_3 + 4u_1 + 6u_2 + u_3 + 2u_4 + 4u_5, \\
F_2 &= u_1^2u_2u_5 + 2u_1^2u_2u_7 + 4u_1u_2u_8 + 4u_1u_2u_9 + u_1u_2u_3 + u_1u_2u_4 + 2u_1u_2u_5 + u_1u_2u_3 + 4u_1 + 6u_2 + u_3 + 2u_4 + 4u_5, \\
F_3 &= 2u_2u_3u_9 + 4u_3u_4 + u_2u_3u_9 + 6u_3 + 5u_4 + 3u_5 + 4, \\
F_4 &= 2u_2u_3u_9 + 4u_3u_4 + u_2u_3u_9 + 6u_3 + 5u_4 + 3u_5 + 3u_8 + 3.
\end{align*}
\tag{25}
\]

Next, the squares have to be reduced using properties of binary variables such that, for any binary variable \( u, u^k = u \). Then, each equation is transformed from the pseudo-Boolean function over \( \mathbb{F}_2 \) to the pseudo-Boolean function over integers [9]. Then, the square of each equation is computed and the sum of all squared equations is determined.

After this step, we have to make quadratization and then add penalties. The other method is first to make linearization of each of the equations and then square each of them, compute their sum, and add penalties [15]. The latter method allows to compute the number of necessary variables more easily, while the former method may result in a smaller number of variables after problem reduction.

Based on these considerations, the first method has been selected, and the final problem in the BQM has been obtained. To make all transformations, the Magma Computational Algebra System (http://magma.maths.usyd.edu.au/magma/) has been used. The task was solved using quantum annealing, with the minimal energy criterion.

The proper solution was found, which is \( y = 7 \), because \( u_1, u_2, u_3 = 1 \). The values of parameters used in solving this QUBO problem are shown in Tab. 2 and the embedding of the problem to the D-Wave Advantage is illustrated in Fig. 3.
Table 2. D-Wave Advantage solver parameters used in solving QUBO problem equivalent to the problem of finding elliptic curve discrete logarithm over $F_7$ on Edwards curve in a subgroup of size 8.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name (chip ID)</td>
<td>Advantage_system 6.1</td>
</tr>
<tr>
<td>Available qubits</td>
<td>5760</td>
</tr>
<tr>
<td>Topology</td>
<td>Pegasus</td>
</tr>
<tr>
<td>Number of reads</td>
<td>10000</td>
</tr>
<tr>
<td>Annealing time</td>
<td>20 µs</td>
</tr>
<tr>
<td>Anneal schedule</td>
<td>$[0,0], [20,1]$</td>
</tr>
<tr>
<td>H gain schedule</td>
<td>$[0,0], [20,1]$</td>
</tr>
<tr>
<td>Programming thermalization</td>
<td>1000 µs</td>
</tr>
<tr>
<td>Number of source variables</td>
<td>58</td>
</tr>
<tr>
<td>Number of target variables</td>
<td>112</td>
</tr>
<tr>
<td>Maximum chain length</td>
<td>6</td>
</tr>
<tr>
<td>Chain strength</td>
<td>15000</td>
</tr>
<tr>
<td>QPU access time</td>
<td>639633.56 µs</td>
</tr>
<tr>
<td>QPU programming time</td>
<td>15233.56 µs</td>
</tr>
<tr>
<td>QPU sampling time</td>
<td>624400 µs</td>
</tr>
<tr>
<td>Total post processing time</td>
<td>10454 µs</td>
</tr>
<tr>
<td>Post processing overhead time</td>
<td>1784 µs</td>
</tr>
</tbody>
</table>

Table 3. Estimated number of logical variables of equivalent QUBO problem for real-world problems.

<table>
<thead>
<tr>
<th>Bitlength of the basefield</th>
<th>Estimated number of the logical variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>6.29 · 10^6</td>
</tr>
<tr>
<td>192</td>
<td>2.12 · 10^7</td>
</tr>
<tr>
<td>256</td>
<td>5.03 · 10^7</td>
</tr>
<tr>
<td>384</td>
<td>1.70 · 10^8</td>
</tr>
<tr>
<td>512</td>
<td>4.03 · 10^8</td>
</tr>
</tbody>
</table>

5. Conclusion

This paper presents methods for transforming the elliptic curve discrete logarithm problem over prime fields to the QUBO problem. We showed how to efficiently perform such a transformation using approximately $3n^3$ logical variables (logical qubits) in the case of the Edwards curve. The discrete logarithm problem in the multiplicative subgroup over a finite field requires approximately $2n^2$ variables.

Table 3 presents the estimated number of logical variables of equivalent QUBO problems for real-world parameters. From Table 3 we can conclude that the number of logical variables (qubits) necessary to run appropriate QUBO problems for real-world parameters is very high. We show that the proposed approach, unlike Shor’s algorithm, may be run in practice.

Even though small instances were solved only, they were run on the D-Wave computer remotely, via the D-Wave Leap cloud (https://cloud.dwavesys.com/leap/).

The elliptic curve discrete logarithm problem on the Edwards curve over $F_7$ has been solved using the D-Wave Advantage QPU, with 112 physical qubits being used. Because the presented approach requires approximately $3n^3$ logical qubits for the reduction of ECDLP over $F_p$, where the bitlength of $p$ is equal to $n$, the number of variables for real-size problems will be very large. For example, for a 256-bit prime field, approximately 50 300 000 logical variables will be necessary.

Even though the problem we have solved is small, according to our knowledge, this is the first instance anyone has ever reported a solution to a discrete logarithm problem over prime fields using direct quantum methods.

Acknowledgments

This work was supported under the Military University of Technology’s University Research Grant no. 858/2021 "Mathematical methods and models in computer science and physics".

References


