


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

Michał Wroński  and Mateusz Leśniak 

NASK - National Research Institute, Warsaw, Poland,
michal.wronski@nask.pl, mateusz.lesniak@nask.pl

Keywords: Quantum annealing · Discrete Logarithm Problem · Binary fields · QUBO · normal bases

Introduction

Shor's algorithm [6] has long been considered the most promising method for solving classical cryptography problems such as integer factorization, the discrete logarithm problem in finite fields, and the elliptic curve discrete logarithm problem.

In 2023, Shor's algorithm was improved by Regev [5], and later by Ragavan and Vaikuntanathan [4] in the case of integer factorization, and then by Ekerå and Gärtner in the case of the discrete logarithm problem in finite fields [2].

However, it is worth noting that other quantum methods have become more prominent with the existing quantum hardware. Taking this into account, quantum annealing is now considered the most practical method of quantum computation, which allows for computing some instances of the discrete logarithm problem (DLP) over finite fields that are larger than those solvable using Shor's algorithm [7], [8].

While computing DLP over prime fields using quantum annealing has been considered before, no author has considered DLP over binary fields using quantum annealing.

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

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

1 DLP in \mathbb{F}_{2^n} using optimal normal bases

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

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

It is worth noting that it is always possible to find such a polynomial $f(t)$ of degree n that $\beta = t$ is the generator of the normal basis in the field \mathbb{F}_{2^n} .

Such an irreducible polynomial can be constructed recursively using Dickson polynomials in the following manner [3]:

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

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

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

In such a case, one can define the multiplication matrix $T^{(0)}$, which allows one to obtain the least significant bit c_0 of the resulting register C . However, by making rotations of columns and rows, it is also possible to obtain other bits of register C [1].

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

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

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

$$\begin{aligned} c_k = & a_{(i_0+k \bmod n)} b_{(j_0+k \bmod n)} \\ & + (a_{(i_1+k \bmod n)} b_{(j_1+k \bmod n)} + a_{(j_1+k \bmod n)} b_{(i_1+k \bmod n)}) \\ & + (a_{(i_2+k \bmod n)} b_{(j_2+k \bmod n)} + a_{(j_2+k \bmod n)} b_{(i_2+k \bmod n)}) \\ & + (a_{(i_3+k \bmod n)} b_{(j_3+k \bmod n)} + a_{(j_3+k \bmod n)} b_{(i_3+k \bmod n)}) \\ & + (a_{(i_4+k \bmod n)} b_{(j_4+k \bmod n)} + a_{(j_4+k \bmod n)} b_{(i_4+k \bmod n)}). \end{aligned} \quad (3)$$

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

We begin the main part of this section by defining the discrete logarithm problem, similarly as in [7, 8]:

$$t^y = h, \quad (4)$$

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

$$t^y \equiv h \pmod{f(t)}, \quad (5)$$

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

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

$$t^y = t^{2^{n-1}u_n} \dots t^{2u_2} t^{u_1}. \quad (6)$$

Let us also note that

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

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

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

$$c_k = a_i u_l + a_j u_l + a_k (1 - u_l). \quad (8)$$

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

2 Example analysis for DLP over \mathbb{F}_{2^5} and its generalization

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

Consider the given field \mathbb{F}_{2^5} . For such a field, there exists an optimal normal basis of type II [1].

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

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

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

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

$$\begin{aligned} c_0 &= a_4b_4 + (a_0b_1 + a_1b_0) + (a_1b_3 + a_3b_1) + (a_2b_4 + a_4b_2) + (a_2b_3 + a_3b_2), \\ c_1 &= a_0b_0 + (a_1b_2 + a_2b_1) + (a_2b_4 + a_4b_2) + (a_3b_0 + a_0b_3) + (a_3b_4 + a_4b_3), \\ c_2 &= a_1b_1 + (a_2b_3 + a_3b_2) + (a_3b_0 + a_0b_3) + (a_4b_1 + a_1b_4) + (a_4b_0 + a_0b_4), \\ c_3 &= a_2b_2 + (a_3b_4 + a_4b_3) + (a_4b_1 + a_1b_4) + (a_0b_2 + a_2b_0) + (a_0b_1 + a_1b_0), \\ c_4 &= a_3b_3 + (a_4b_0 + a_0b_4) + (a_0b_2 + a_2b_0) + (a_1b_3 + a_3b_1) + (a_1b_2 + a_2b_1). \end{aligned} \quad (10)$$

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

For any $A \in \mathbb{F}_{2^5}$ given as $a_4t^4 + a_3t^3 + a_2t^2 + a_1t^{2^1} + a_0t^{2^0}$, let us set, for example, $B = t^{2^0u_0}$. In such a case, the result C of the multiplication of $A \cdot B$, will be given as:

$$\begin{aligned} c_0 &= a_1u_0 + a_0(1 + u_0), \\ c_1 &= a_0u_0 + a_3u_0 + a_1(1 + u_0), \\ c_2 &= a_3u_0 + a_4u_0 + a_2(1 + u_0), \\ c_3 &= a_2u_0 + a_1u_0 + a_3(1 + u_0), \\ c_4 &= a_4u_0 + a_2u_0 + a_4(1 + u_0). \end{aligned} \quad (11)$$

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

Therefore, we may estimate the total number of variables required for transforming DLP over \mathbb{F}_{2^n} to the QUBO problem. We use the same decomposition tree as in [7].

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

However, the amount of variables described above ($4n^2$) may be lowered. Let us note that each of the single equations c_i can be transformed into a pseudo-boolean function in the following manner (let us take, for example, c_1): $c_1 + a_0u_0 + a_3u_0 + a_1(1 + u_0) = 0$.

As the maximal value of the left side of the equation is equal to 5, two new variables are necessary for k . However, note that in binary notation, the equation above is equivalent to $-c_1 + a_0u_0 + a_3u_0 + a_1(1 - u_0) = 0$. Now one can transform this equation into the pseudo-boolean function as $-c_1 + a_0u_0 + a_3u_0 + a_1(1 - u_0) - 2k_1 = 0$, where $k_1 \in \{0, 1\}$. Why is it possible? Let's observe that the maximal value of $-c_1 + a_0u_0 + a_3u_0 + a_1(1 - u_0)$ is equal to 2, and the minimal value is equal to -1 . It means that $0 \leq k_1 \leq 1$. Therefore, we require only one bit for multiplicity representation instead of two bits, as was described above. It is important to see that a similar trick may be done for all fields \mathbb{F}_{2^n} , for which optimal normal bases exist. In such a case, transforming the DLP problem over \mathbb{F}_{2^n} will require $3n^2$ logical variables. However, whether this transformation may be obtained using fewer variables is unknown.

3 Working example

We conducted a practical experiment to solve the discrete logarithm problem in \mathbb{F}_{2^3} , given by

$$t^y \equiv t^4 + t^2 \pmod{(t^3 + t^2 + 1)}. \quad (12)$$

We transformed the problem above into the QUBO problem using 11 logical and 14 physical variables. We obtained the correct solution of $y = 5$ using quantum annealing.

The full description of this example is available here https://github.com/wronskimichal/Example-QA-for-DLP-in-mathbb-F_-2-3-field/blob/main/CECC_2024_QA_DLP.pdf

References

- [1] Alaaeldin Amin and Turki Faisal Al-Somani. Hardware implementations of $gf(2^m)$ arithmetic using normal basis. *Journal of Applied Sciences*, 6(6):1362–1372, 2006.
- [2] Martin Ekerå and Joel Gärtner. Extending regev's factoring algorithm to compute discrete logarithms, 2023.
- [3] Ronald C Mullin and Ayan Mahalanobis. *Dickson bases and finite fields*. Faculty of Mathematics, University of Waterloo, 2005.
- [4] Seyoon Ragavan and Vinod Vaikuntanathan. Optimizing space in regev's factoring algorithm. Cryptology ePrint Archive, Paper 2023/1501, 2023.
- [5] Oded Regev. An efficient quantum factoring algorithm, 2023.
- [6] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.
- [7] Michał Wroński. Practical solving of discrete logarithm problem over prime fields using quantum annealing. In *Computational Science – ICCS 2022*, pages 93–106, Cham, 2022. Springer International Publishing.
- [8] Michał Wroński, Elżbieta Burek, Łukasz Dzierzkowski, and Olgierd Żołnierczyk. Transformation of elliptic curve discrete logarithm problem to qubo using direct method in quantum annealing applications. *Journal of Telecommunications and Information Technology*, (1):75–82, 2024.