이학박사 학위논문

Discrete Logarithm Problem with Auxiliary Inputs

(부가정보를 이용한 이산대수 문제 연구)

2014년 2월

서울대학교 대학원

수리과학부

김태찬

Discrete Logarithm Problem with Auxiliary Inputs

(부가정보를 이용한 이산대수 문제 연구)

지도교수 천정희

이 논문을 이학박사 학위논문으로 제출함

2013년 11월

서울대학교 대학원

수리과학부

김태찬

김태찬의 이학박사 학위논문을 인준함

2013년 12월

위 원 장	김 명 환	(인)
부위원장	천 정 희	(인)
위 원	이 향 숙	(인)
위 원	변 동 호	(인)
위 원	오 병 권	(인)

Discrete Logarithm Problem with Auxiliary Inputs

A dissertation

submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

to the faculty of the Graduate School of
Seoul National University

by

Taechan Kim

Dissertation Director : Professor Jung Hee Cheon

Department of Mathematical Sciences
Seoul National University

February 2014

Abstract

The modern cryptography has been developed based on mathematical hard problems. For example, it is considered hard to solve the discrete logarithm problem (DLP). The DLP is required to solve α for given g, g^α, where $G = \langle g \rangle$. It is well-known that the lower bound complexity to solve the DLP in the generic group model is $\Omega(p^{1/2})$ (EUROCRYPT 97, Shoup), where p is the prime order of the group G. However, if the problem is given with auxiliary informations, then it can be solved faster than $O(p^{1/2})$. In the former of the thesis, we deal with the problem called discrete logarithm problem with the auxiliary inputs (DLPwAI). The DLPwAI is a problem required to solve α for given $g, g^\alpha, \ldots, g^{\alpha^d}$. The state-of-art algorithm to solve this problem is Cheon's algorithm which solves the problem in the case of $d|p \pm 1$.

In the thesis, we propose a new method to solve the DLPwAI which reduces to find a polynomial with small value sets. As a result, we solved the DLPwAI when g^{α^k} were given, where k is an element of multiplicative subgroup of $\mathbb{Z}_{p-1}^\times$.

In the later of the thesis, we try to solve the DLP with the pairing inversion problem. If one has an efficient algorithm to solve the pairing inversion, then it can be used to solve the DLP. We focus on how to reduce the complexity of the pairing inversion problem by reducing the size of the final exponentiation in the pairing computation. As a result, we obtained the lower bound of the size of the final exponentiation.

Key words: discrete logarithm problem, pairing inversion, Cheon's algorithm, Dickson polynomial

Student Number: 2007-20270

Contents

Chapter 1

Introduction

In the thesis, we try to solve the discrete logarithm problem with auxiliary inputs and the pairing inversion problem. These problems play a staple role in the cryptography since their hardness supports the security of many cryptosystems.

The discrete logarithm problem (DLP) is asked to compute $\alpha \in \mathbb{F}_p$ for given g and g^α, where g is a generator of a group of prime order p. The DLP with auxiliary inputs (DLPwAI) is the problem to compute $\alpha \in \mathbb{F}_p$ for given $g, g^\alpha, \ldots, g^{\alpha^d}$. Certainly, it is seemingly easier the DLPwAI than the DLP since it is given more hints. The generic lower bound of the complexity of the DLPwAI is smaller than the DLP by a factor $\sqrt{d}$ for $d < p^{1/3}$. This problem is widely used to construct many cryptosystems with various functionalities, though it has potential weakness.

The first algorithm to solve the DLPwAI was given by Cheon [11, 12] for $p\pm1$ cases. For $p-1$ case, it is solved independently by Brown and Gallant [8]. Since then, several generalizations to the $\Phi_k(p)$ cases were given [35, 47] following Cheon's approach.

We consider the different approach to solve the DLPwAI. The approach is to use the polynomial with the small value sets. Our observation leads us to consider the Dickson polynomial and its generalization. However, the practicality of this generalization is still remained open.

On the line of the research, we also consider the generalized version of the DLPwAI (GDLPwAI) which is a problem to compute $\alpha \in \mathbb{F}_p$ for given $g^{\alpha^{e_i}}$ for $i = 1, \ldots, d$. Our research gives a method to solve the GDLPwAI where e_i's forms a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$.

Finally, we consider the pairing inversion problem. A pairing is a non-degenerate bilinear map $e : G_1 \times G_2 \to G_T$. The pairing inversion problem is to compute P (or Q) where Q (or P) and $e(P, Q)$ were given. It is easy to solve the computational Diffie-Hellman problem when efficient pairing inversion algorithm exists.

Mostly used pairing in the cryptography is the Tate pairing. The Tate pairing is computed by the Miller step and the final exponentiation step in the given order. Thus the inversion is followed by the exponent inversion and the Miller inversion. Since the recent results [33, 10] show that the pairing inversion reduces to the exponent inversion, we only consider reducing the complexity of the final exponentiation in the Tate pairing. Our result gives a universal approach to reduce the final exponentiation and shows that the value is the lower bound.

Contributions

The thesis contains a joint work with Jung Hee Cheon and Yong Soo Song [14] which appears in Selected Areas in Cryptography 2013 and a work with Sungwook Kim and Jung Hee Cheon [38] which appears in IEEE transactions on Information Theory. It also includes a prepublication with Jung Hee Cheon [37]. A part of the article will also appear in a Chapter of the proceeding in deGruyter [13].

Chapter 2

Discrete Logarithm Problem

2.1 Algorithms for the DLP

In this section, we describe well-known algorithms to solve the DLP. Since the thesis mainly deals with the DLPwAI, we leave aside more details on basic DLP algorithms referring to [21].

2.1.1 Generic algorithms

Consider a cyclic group G of order n which is not necessarily prime. A generic algorithm takes as input n and encodings of group elements. It is also given access to oracle that returns encoding of binary operation or inversion of the given group elements. In the sense of the generic group model [51], the lower bound of the complexity solving the DLP is $\Omega(\sqrt{n})$.

The Baby-Step Giant-Step (BSGS) technique is an algorithm to solve the DLP deterministically in $O(\sqrt{n})$ time complexity. It computes α by making two lists of elements of G and finding a collision.

Pohlig-Hellman algorithm solves the DLP efficiently when the order n has only small prime factors. For $n = \prod_i q_i^{f_i}$, the algorithm in advance solves

$\alpha \mod q_i^{f_i}$ for each i using the BSGS technique, and then recovers α from the Chinese Remainder Theorem (CRT). So, the total complexity depends on the size of the largest prime factor of n.

While the BSGS technique requires the $O(\sqrt{n})$ storage when it makes the lists, the Pollard's rho algorithm only requires the storage of the constant size although it is probabilistic. The Pollard's kangaroo algorithm is also probabilistic algorithm which solves the discrete logarithm α contained in specific interval $[a, b]$. Although the Pollard's rho algorithm is more efficient than running the Pollard's kangaroo algorithm for the entire interval $[0, n)$, it is efficient when the interval is small.

These algorithms attain the generic lower bound complexity, though they still have the exponential time complexity.

Baby-Step Giant-Step

The BSGS technique is a simple, generic and deterministic algorithm to solve the DLP. The total complexity is $O(\sqrt{n})$ exponentiations in G, it also needs to store $O(\sqrt{n})$-number of elements of G.

For given g and $h = g^\alpha$, compute two lists

$$L_1 = \{g^{-i}h : 0 \le i \le \lfloor\sqrt{n}\rfloor\} \quad \text{and} \quad L_2 = \{g^{\lceil\sqrt{n}\rceil j} : 0 \le j \le \lfloor\sqrt{n}\rfloor\},$$

then compare elements of L_1 and L_2. If a collision $g^{-i_0}h = g^{\lceil\sqrt{n}\rceil j_0}$ occurs, then the discrete logarithm is calculated from $h = g^{i_0+\lceil\sqrt{n}\rceil j_0}$ and $\alpha = i_0 + \lceil\sqrt{n}\rceil j_0$. To show the existence of a collision, take two integers $j_0 = \lfloor\frac{\alpha}{\lceil\sqrt{n}\rceil}\rfloor$ and $i_0 = \alpha - \lceil\sqrt{n}\rceil j_0$ for $0 \le \alpha < n$, and check $0 \le i_0, j_0 \le \lfloor\sqrt{n}\rfloor$ and $\alpha = i_0 + \lceil\sqrt{n}\rceil j_0$. Therefore, two lists L_1 and L_2 always have a common element. In fact, the elements of the list L_1 need not to be stored. Precomputing and storing the list L_2, collision finding can be done by computing and looking up an element of L_1 with elements L_2. Note that the list L_2 may be used to solve the DLP for another element h' of G.

The Pohlig-Hellman algorithm

If all prime factors of an integer n is less than a positive real number B, then n is called B-smooth. The Pohlig-Hellman algorithm solves the DLP deterministically when n is a smooth number.

Let P be the set of prime divisors of n, and $n = \prod_{q \in P} q^{e_q}$ be the factorization. The main idea of the Pohlig-Hellman algorithm is to compute $\alpha \pmod{q^{e_q}}$ for each $q \in P$ for $\alpha = \log_g h$. Then one can efficiently recover $\alpha \in \mathbb{Z}_n$ by the Chinese Remainder Theorem (CRT).

Consider a prime divisor $q \in P$. There exist $c_0, c_1, \ldots, c_{e_q-1} \in [0, q)$ satisfying $\alpha \equiv c_0 + c_1 q + \cdots + c_{e_q-1} q^{e_q-1} \pmod{q^{e_q}}$. The coefficients $c_0, c_1, \ldots, c_{e_q-1}$ are determined inductively as follows. First, from the equations $\alpha \equiv c_0 \pmod{q}$ and $\left(g^{\frac{p-1}{q}}\right)^{c_0} = h^{\frac{p-1}{q}}$, one computes c_0 in $O\left(\sqrt{q}\right)$ using the BSGS technique. Note that two elements $g^{\frac{p-1}{q}}$ and $h^{\frac{p-1}{q}}$ are contained in $H = \langle g^{\frac{p-1}{q}} \rangle$, which is a subgroup of G of prime order q. Therefore, $c_0 \in [0, q)$ is uniquely determined. Inductively, the next coefficient c_i is obtained from the equations $\alpha \equiv c_0 + c_1 q + \cdots + c_i q^i \pmod{q^{i+1}}$ and $g^{(c_0 + c_1 q + \cdots + c_i q^i)\frac{p-1}{q^{i+1}}} = h^{\frac{p-1}{q^{i+1}}}$, which is equivalent to $\left(g^{\frac{p-1}{q}}\right)^{c_i} = g^{-(c_0 + c_1 q + \cdots + c_{i-1} q^{i-1})\frac{p-1}{q^{i+1}}} h^{\frac{p-1}{q^{i+1}}}$. It is done in $O\left(\sqrt{q}\right)$ exponentiations using the BSGS. Repeating this process for all $q \in P$, every modulus $\alpha \pmod{q^{e_q}}$ is obtained in $O\left(\sum_{q \in P} e_q \sqrt{q}\right)$ exponentiations, and $\alpha \in \mathbb{Z}_n$ is recovered from them.

Pollard's rho algorithm

The BSGS technique requires $O\left(\sqrt{n}\right)$ memory. The Pollard's rho algorithm is one way to overcome storage.

For given g and $h = g^{\alpha}$, the Pollard's rho algorithm uses a function $f : G \to G$, where G is partitioned into three sets S_0, S_1, S_2 with roughly same sizes. The function f is constructed in a way that the exponents of g and h are traceable, precisely, it should be easy to compute $(x_{i+1}, \beta_{i+1}, \gamma_{i+1})$ from (x_i, β_i, γ_i) for $x_{i+1} := f(x_i)$ and $x_i = g^{\beta_i} h^{\gamma_i}$. The typical example of the function $f(x)$ is as follows:

$$x_{i+1} := f(x_i) = \begin{cases} hx_i, & x_i \in S_0 \\ x_i^2, & x_i \in S_1 \\ gx_i, & x_i \in S_2 \end{cases}$$

In this case, the exponents β_i and γ_i are traceable in the following ways:

$$\beta_{i+1} = \begin{cases} \beta_i, & x_i \in S_0 \\ 2\beta_i, & x_i \in S_1 \\ \beta_i + 1, & x_i \in S_2 \end{cases} \quad \text{and} \quad \gamma_{i+1} = \begin{cases} \gamma_i + 1, & x_i \in S_0 \\ 2\gamma_i, & x_i \in S_1 \\ \gamma_i, & x_i \in S_2 \end{cases}$$

Since G is a finite set, the sequence $\{x_1, x_2, \dots\}$ obtained by evaluating the function f iteratively must contains a cycle. Using the Floyd's cycle detection algorithm, a collision $x_i = x_{2i}$ finds a discrete logarithm with the storage of the constant size under the assumption that f looks like a random function.

The r-adding walk method is a generalized version of the Pollard's rho algorithm that uses a function with G partitioned into r disjoint sets. It is known that the 20-adding walk is very close to the random walk [52].

Pollard's kangaroo algorithm

Pollard's kangaroo algorithm solves the DLP when the discrete logarithm $\alpha \in [0, n)$ is contained in a certain interval $[a, b]$. The choice $a = 0, b = n - 1$ for entire α is possible, but Pollard's rho algorithm is more efficient in this case.

One precomputes $g^{e_i}, 1 \leq i \leq r$ for some small integers $e_1, \cdots, e_r$ whose sizes are $\sqrt{b-a}$ approximately. Let $f : G \to \{1, 2, \cdots, r\}$ be a pseudorandom function. For a suitable integer N, compute x_N as follows

$$x_0 = g^b, \ x_{i+1} = x_i g^{e_{f(x_i)}} \ \text{ for } \ i = 0, 1, \ldots, N-1.$$

Then until a collision $y_j = x_N$ is detected, compute the followings

$$y_0 = h, \ y_{j+1} = y_j g^{e_{f(y_j)}} \ \text{ for } \ j = 0, 1, \ldots, N-1.$$

The sequence $\{x_0, x_1, \ldots\}$ is called a tame kangaroo and $\{y_0, y_1, \ldots\}$ a wild kangaroo. Since the mean step size is $m = (\sum_{i=1}^{r} e_i)/r \approx \sqrt{b-a}$, the wild kangaroo meets the tame kangaroo with probability $1/m$. The complexity of the algorithm becomes $O(\sqrt{b-a})$.

2.1.2 Non-generic algorithms

In this subsection, we recall non-generic algorithms solving the DLP which can be used only in specific groups such as $\mathbb{Z}_p^*$ or $\mathbb{F}_q^*$ for prime power q. These algorithms exploits the specifications of the group structures bringing more efficiency than the generic algorithms.

The index calculus is an efficient way to solve the DLP when $G = \mathbb{Z}_p^*$. It consists of two steps: sieving and decent. In the sieving phase, it precomputes the discrete logarithms of the factor base, usually a set of small primes, by finding sufficiently many relations. In the decent phase, one computes the discrete logarithm of arbitrarily given element. This algorithm runs in subexponential time. The idea of the original index calculus is improved to the number field sieve and function field sieve algorithms [1, 24, 25, 31]. In particular, the complexity becomes quasi-polynomial time when the characteristic p is small [1, 24].

Index calculus

Consider the index calculus algorithm over a multiplicative group $G = \mathbb{Z}_p^*$. The index calculus algorithm is a probabilistic algorithm based on the prime factorization of integers. Suppose that g is a fixed generator of G. Taking a suitable bound B, let $q_0 = -1$, and $q_1 = 2 < q_2 = 3 < \cdots < q_d$ be the primes less than B. One precomputes the discrete logarithm problems of the factor base q_i as follows: for randomly chosen $\beta \in \mathbb{Z}_{p-1}$, one computes the factorization of g^β modulo p. If $g^\beta = \prod_{i=0}^d q_i^{e_i}$ is a B-smooth number, we have an equation $\beta = e_0\beta_0 + \cdots + e_d\beta_d$ in $\mathbb{Z}_{p-1}$, if g^β was not a B-smooth number then try it again for another β. Repeating this process many times, we obtain $d+1$ number of linearly independent equations. Then the discrete logarithms of q_i are recovered from the linear algebra.

Now, for given $h = g^\alpha$, we choose a random element $\gamma \in \mathbb{Z}_{p-1}$ repeatedly until $hg^\gamma \pmod p$ is expressed as a product of primes less than B. If we find a such γ, then α is determined by $hg^\gamma = \prod_{i=0}^d q_i^{f_i}$ and $\alpha = -\gamma + \sum_{i=0}^d f_i\beta_i$. The expected complexity is $L_n[1/2, \sqrt{2} + o(1)]$ for a suitable bound B. Here, L-notation is defined as

$$L_p[\theta, c] = \exp\left[(c + o(1))(\log p)^\theta(\log\log p)^{1-\theta}\right]$$

for $c > 0$ and $0 \leq \theta \leq 1$. Note that L_p is a polynomial function of $\log p$ when $\theta = 0$, and an exponential function of $\log p$ when $\theta = 1$. The total complexity $L_p[1/2, \sqrt{2} + o(1)]$ of the index calculus is a subexponential function of $\log p$.

Chapter 3

Discrete Logairhtm Problem with Auxiliary Inputs

3.1 Introduction

In the recent decades, many variants of the DLP such as Weak Diffie-Hellman Problem (WDHP) [42], Strong Diffie-Hellman Problem (SDHP) [5], Bilinear Diffie-Hellman Inversion Problem (BDHIP) [4] and Bilinear Diffie-Hellman Exponent Problem (BDHEP) [7] are introduced to guarantee the security of many cryptosystems such as the traitor tracing [42], the short signatures [5], ID-based encryption [4], the broadcast encryption [7] and so on. The instants of these problems contain additional information more than the DLP. These problems are widely used since they enable the construction of the cryptosystems with various functionalities, though such auxiliary information could weaken the problems.

The first realization of the weakness of these problems is done by Cheon [11, 12] and by Brown and Gallant [8] independently. Throughout the thesis, we mainly follow the notations from Cheon's algorithm. Cheon realized that the problems can be considered as the problem which solves α when $g, g^{\alpha}, \cdots, g^{\alpha^d}$ are given and called this problem by Discrete Logarithm Problem with Auxiliary Inputs (DLPwAI). The DLPwAI can be solved efficiently in time complexity $O(\sqrt{p/d})$ when d is a small divisor of $p \pm 1$ and p is prime order of the group G. This complexity is the same with the lower bound for the DLPwAI in the generic group model [51]. Since the lower bound for the original DLP is $O(\sqrt{p})$ in the generic group model, Cheon's algorithm shows the evidence of the weakness of DLPwAI in some cases.

The idea of Cheon's algorithm is to embed the discrete logarithm α into the finite fields $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. Precisely, he exploits the fact that α^d can be embedded into an element of the small subgroup of $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$ when d is a divisor of $p \pm 1$. After Cheon's algorithm, Satoh [47] generalized this algorithm using the embedding of $\alpha \in \mathbb{F}_p$ into the general linear group $GL_k(\mathbb{F}_p)$. The generalization tried to solve the problem when d is a divisor of $\Phi_k(p)$ for the k-th cyclotomic polynomial $\Phi_k(\cdot)$, but the complexity for $k \geq 3$ was not clearly understood. Recently, Kim et al. [35] simply realized that Satoh's generalization is essentially the same with the embedding of $\mathbb{F}_p$ into $\mathbb{F}_{p^k}$ and clarified the complexity of the algorithm. Unfortunately, their result says that in most cases the complexity of this generalization is not faster than the current square root complexity algorithm such as Pollard's rho algorithm [45] for $k \geq 3$.

All the above algorithms use the embedding technique of the finite field which can be considered as the quantitative version of the reduction algorithms from DLP into Diffie-Hellman problem [39, 40]. On the other hand, we propose an algorithm to solve the DLPwAI with the polynomial mapping instead of the embedding of the element. The idea is to choose a polynomial f of degree d and compute two lists of $g^{f(r_i \alpha)}$ and $g^{f(s_j)}$ for random elements

r_i and s_j using the fast multipoint evaluation, and find a collision between them. For the efficiency of this approach, we should consider three things:

1. how to compute $g^{f(r_i\alpha)}$ efficiently for given $g, g^{\alpha}, \ldots, g^{\alpha^d}$ (Section 3.3),
2. how many to choose random r_i and s_j for a collision (Section 3.4), and
3. how to choose a polynomial f (Section 3.5).

We begin with the description of the previous works, Cheons' algorithm. This chapter includes a part of the prepublicated work [37] with Jung Hee Cheon and the survey article to appear in deGruyter proceedings [13].

Organization This chapter is organized as follows: we recall the DLPwAI and Cheon's algorithm in Section 3.2. Several trials to generalize the Cheon's algorithm are also contained in this section. We explain our approach to solve the DLPwAI using polynomials in Section 3.5.

3.2 The DLPwAI and Cheon's algorithm

The DLPwAI requires to solve $\alpha \in \mathbb{Z}_p$ for given $g, g^{\alpha}, \ldots, g^{\alpha^d}$. In the generic group model [51], the lower bound of the complexity solving this problem is $O(\sqrt{p/d})$ when $d < p^{1/3}$. It is less than $O(\sqrt{p})$ which is the generic lower bound of the DLP. There are generic algorithms for the DLP achieving the lower bound complexity, however, for the DLPwAI, only Cheon's algorithm achieves the lower bound in a few cases.

3.2.1 $p-1$ cases

Assume that three elements $g, g_1 = g^{\alpha}$ and $g_d = g^{\alpha^d}$ are given for a divisor d of $p-1$. The main idea of Cheon's algorithm is to exploit the fact that α^d is contained in the subgroup of $\mathbb{Z}_p^*$ of small order $\frac{p-1}{d}$. By applying the BSGS technique on this smaller group, one can recover α^d. Then α is recovered in a similar fashion.

To start Cheon's algorithm, we choose a primitive element ξ of $\mathbb{Z}_p$. Since $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$, there are exactly $\phi(p-1)$-number of primitive elements in $\mathbb{Z}_p$. For a randomly chosen element in $\mathbb{Z}_p^*$, it is a primitive element with the probability $\frac{\phi(p-1)}{p-1} \geq \frac{1}{6\log\log(p-1)}$, which is sufficiently large. So it may be assumed that a primitive element ξ of $\mathbb{Z}_p$ can be found efficiently.

Theorem 3.2.1 ([12])**.** *Let d be a divisor of $p-1$. For given $g, g_1 = g^\alpha$ and $g_d = g^{\alpha^d}$, one can solve α deterministically in $O\left(\sqrt{\frac{p-1}{d}} + \sqrt{d}\right)$ exponentiations with the storage $O\left(\max\{\sqrt{\frac{p-1}{d}}, \sqrt{d}\}\right)$.*

Proof. Consider a primitive element ξ of $\mathbb{Z}_p$. Define $\zeta = \xi^d$ and $m = \lceil\sqrt{\frac{p-1}{d}}\rceil$. There exist two integers $k_1 \in [0, d)$ and $k_2 \in [0, \frac{p-1}{d})$ such that $\alpha = \xi^{\frac{p-1}{d}k_1 + k_2}$. We will calculate k_1 and k_2 using two independent BSGS techniques.

First, we find k_2 using the BSGS technique. From $\alpha^d = \xi^{dk_2} = \zeta^{k_2}$ and $g_d = g^{\alpha^d} = g^{\zeta^{k_2}}$, there exist two integers $0 \leq u_2, v_2 \leq \lfloor\sqrt{\frac{p-1}{d}}\rfloor$ such that $k_2 = mu_2 + v_2$, or equivalently $\alpha^d\zeta^{-v_2} = \zeta^{mu_2}$ and $g_d^{\zeta^{-v_2}} = g^{\zeta^{mu_2}}$. Two integers u_2 and v_2 are determined in $O\left(\sqrt{\frac{p-1}{d}}\right)$ exponentiations. After finding k_2, we again use the BSGS technique similarly, and determine k_1 in $O(\sqrt{d})$ exponentiations from the equation $g_1 = g^\alpha = g^{\xi^{\frac{p-1}{d}k_1+k_2}}$. The total complexity is $O\left(\sqrt{\frac{p-1}{d}} + \sqrt{d}\right)$ exponentiations with $O\left(\max\{\sqrt{\frac{p-1}{d}}, \sqrt{d}\}\right)$ storage of elements of G. $\square$

Note that the total complexity $O\left(\max\{\sqrt{\frac{p-1}{d}}, \sqrt{d}\}\right)$ of Cheon's $p-1$ algorithm can be lowered down to $O\left(p^{1/4}\right)$ when $d \approx \sqrt{p}$. Based on Pollard's kangaroo algorithm, it can be run with less storage [12].

He was also able to find α from $g, g^{\alpha}, \cdots, g^{\alpha^{2d}}$ when d is a divisor of $p+1$ using a quadratic extension of $\mathbb{F}_p$.

3.2.2 Generalized algorithms

The idea of Cheon's algorithm is to embed an element in $\mathbb{F}_p$ to an element of an extension field of $\mathbb{F}_p$. More precisely, the discrete logarithm $\alpha \in \mathbb{F}_p$ is embedded into an element in $\mathbb{F}_p$ in $\Phi_1(p) = p - 1$ case. Cheon's algorithm is efficient when $p-1$ has a small divisor d with given parameters $g, g^{\alpha}, \ldots, g^{\alpha^d}$.

Satoh [47] extended Cheon's algorithm into the cases of $\Phi_k(p)$ for $k \geq 3$ by using the embedding of $\mathbb{F}_p$ into $GL(k, \mathbb{F}_p)$. Recently, Kim et al. [35] realized that the Satoh's embedding is essentially the same with the embedding of $\mathbb{F}_p$ into $\mathbb{F}_{p^k}$ and showed that in most cases this generalization cannot be faster than the square-root complexity algorithms such as Pollard's rho algorithm when $k \geq 3$.

Satoh's generalization

The main idea of Cheon's $p + 1$ algorithm is to construct an embedding of $\mathbb{F}_p$ into its quadratic extension $\mathbb{F}_p[\theta]$. Satoh tried to generalize the Cheon's algorithm using an embedding of $\mathbb{F}_p$ into general linear group $GL(k, \mathbb{F}_p)$.

Definition 3.2.1. For a given positive integer ν, define the p-norm $\|\nu\|_p$ by the sum of ν_i's, where ν_i's are integers satisfying $0 \leq \nu_i < p$ and $\nu = \sum_{i \leq 0} \nu_i p^i$.

For a divisor d of $\Phi_k(p)$ for some $k \geq 1$, we put $D := \Phi_k(p)/d$. Satoh's algorithm solves the DLP with inputs $g, g^{\alpha}, \cdots, g^{\alpha^d}$ if it is possible to find an integer u satisfying $\gcd(u, p^k - 1) = 1$ and $u(p^k - 1)/D \equiv \Delta - \delta \pmod{p^k - 1}$, where Δ and δ are integers with small p-norms. The total complexity is in the following theorem.

Theorem 3.2.2 ([47])**.** *Suppose that d is a divisor of $\Phi_k(p)$ for some $k \geq 1$. Moreover, assume that an integer u satisfies* $\gcd(u, p^k - 1) = 1$ *and* $u(p^k - 1)/D \equiv \Delta - \delta \pmod{p^k - 1}$ *for some integers Δ and δ. Then one can solve the DLPwAI in* $\tilde{O}\left(k^2(k \log p + w + k^3 + \sqrt{D})\right)$*, where* $w = \|\Delta\|_p + \|\delta\|_p$.

This theorem is rather complicated to understand the efficiency. Kim et al.'s generalization in the next section covers all cases of Satoh's algorithm, while it uses simpler notations. Moreover, they observed that the generalization of Cheon's algorithm is not so faster than the usual DL-solving algorithm in most cases.

Kim et al.'s generalization

Let $D = \Phi_k(p)/d$ and r be an integer. Kim et al. [35] considered an embedding

$$\mathbb{F}_p \to \mathbb{F}_{p^k}, \quad \alpha \mapsto (\alpha + \theta)^{r(p^k-1)/D},$$

for an element $\theta \in \mathbb{F}_{p^k}^{\times}$ which is not in a proper subfield and they noticed that Satoh's embedding of $\mathbb{F}_p$ into general linear group $GL(k, \mathbb{F}_p)$ is essentially the same with the above embedding when $r = 1$. The element $(\alpha + \theta)^{r(p^k-1)/D}$ is an element of the subgroup of $\mathbb{F}_{p^k}$ of order D, so the idea of Cheon's algorithm can be applied.

Define $E := (p^k - 1)/D$ and write rE in a signed p-ary representation as $rE = \sum_i e_i p^i$, where $|e_i| < p/2$. For an integer $\nu = \sum_i \nu_i p^i$ with the signed representation, a signed sum of digits is $S_p(\nu) := \max\{S_p^+(\nu), S_p^-(\nu)\} = \max\{\sum_{\nu_i > 0} \nu_i, -\sum_{\nu_i < 0} \nu_i\}$.

Consider the followings:

$$(\alpha+\theta)^{rE} = \frac{(\alpha+\theta)^{\sum_{e_i>0} e_i p^i}}{(\alpha+\theta)^{\sum_{e_i<0} |e_i| p^i}} = \frac{\prod_{e_i>0}(\alpha+\theta^{p^i})^{e_i}}{\prod_{e_i<0}(\alpha+\theta^{p^i})^{|e_i|}} = \frac{f_1(\alpha)\theta_1+\cdots+f_k(\alpha)\theta_k}{h_1(\alpha)\theta_1+\cdots+h_k(\alpha)\theta_k},$$

where $\{\theta_1,\dots,\theta_k\}$ is a basis of $\mathbb{F}_{p^k}$ for $\theta_i = \theta^{i-1}$, $\deg f_i \le S_p^+(rE)$ and $\deg h_j \le S_p^-(rE)$. Since this element is in the subgroup of order D, choose a generator ζ of this group and then apply the BSGS technique to find the integer $k \in [0, D)$ satisfying $(\alpha+\theta)^{rE} = \zeta^k$.

The total complexity of this algorithm is about $O\left(\sqrt{D} + S_p(rE)\right)$. Hence, to reduce the total complexity, it is needed to find an integer r such that rE has a low signed weight. However, by [35, Theorem 4.5], this complexity is worse than the ordinary DL solving algorithms unless all prime divisors of D are divisors of k or $p \pm 1$.

When $k = 2$, the complexity of this algorithm is meaningful. When d is a divisor of $\Phi_2(p) = p+1$, put $D = (p+1)/d$ and $E = (p-1)d$. The signed weight of $E = dp - d$ is equal to d, which is sufficiently small. It corresponds to the case $r = 1$ of the above algorithm. Therefore, one can solve the DLPwAI in $O\left(\sqrt{\frac{p+1}{d}} + d\right)$ exponentiations with storage $O\left(\max\{\frac{p+1}{d}, \sqrt{d}\}\right)$ when d is a divisor of $p+1$, and $g, g^{\alpha}, \cdots, g^{\alpha^d}$ are given. Note that the total complexity $O\left(\sqrt{\frac{p+1}{d}} + d\right)$ can be lowered down to $O(p^{1/3})$ when $d \approx p^{1/3}$.

3.3 Fast multipoint evaluation in the black-box manner

Let $f(x)$ be a polynomial over a field $\mathbb{F}$ of degree d, then it is well known that one can compute $f(r_1), \dots, f(r_d)$ in $\tilde{O}(d)$ field operations using the fast mulitipoint evaluation method. The fast multipoint evaluation method follows from the fast multiplication methods, the fast Fourier transformations, the fast polynomial divisions and so on. In this section, we shall show that the fast multipoint evaluation is possible even when the polynomial $f(x)$ is given in the exponentiated form. In other words, we give a fast multipoint evaluation method when $g^{a_d}, \ldots, g^{a_0}$ is given for $f(x) = a_d x^d + \cdots + a_1 x + a_0$. This will be used in next sections to propose another approach to solve the DLPwAI. Precisely, we shall show the followings.

Proposition 3.3.1. *We are given $g^{a_0}, \ldots, g^{a_d}$ for a polynomial $f(x) = a_d x^d + \cdots + a_1 x + a_0 \in \mathbb{F}_p[x]$. One can compute $g^{f(r_1)}, \ldots, g^{f(r_d)}$ in $O(d \log d \log\log d)$ group operations, where $r_1, \ldots, r_d$ are random elements from $\mathbb{F}_p$.*

In this thesis, the group operations include the exponentiations and the multiplications in the group.

The following corollary will be useful throughout this chapter.

Corollary 3.3.1. *For given $g, g^{\alpha}, \ldots, g^{\alpha^d}$ and a polynomial $f(x) \in \mathbb{F}_p[x]$ of degree d, we can compute $g^{f(r_1\alpha)}, \ldots, g^{f(r_d\alpha)}$ in $O(d \log d \log\log d)$ group operations.*

Proof. We can obtain $g^{a_0}, (g^{\alpha})^{a_1}, \ldots, (g^{\alpha^d})^{a_d}$ with d exponentiations from $g, g^{\alpha}, \ldots, g^{\alpha^d}$ and $f(x)$. Let $h(x) := f(x\alpha) = (a_d\alpha^d)x^d + \cdots + (a_1\alpha)x + a_0$ and apply Proposition 3.3.1. $\square$

The proof of Proposition 3.3.1 easily comes from the original method of the fast multipoint evaluation. The main observation is that the multiplication/addition/subtraction in $\mathbb{F}_p$ replaces with the exponentiation/multiplication/division in $\mathbb{G}$. This section mainly refers to [56].

We begin with the description of the fast multiplication algorithm with the Discrete Fourier Transform (DFT) in the blackbox manner. Let ω be a d-th primitive root of unity. The $DFT_{\omega} : \mathbb{F}_p[x] \to \mathbb{F}^d$ is a map given by $\big(f(1), f(\omega), f(\omega^2), \ldots, f(\omega^{d-1})\big)$. The fast blackbox Fourier transform is the algorithm to compute $g^{DFT_\omega(f)} := (g^{f(1)}, g^{f(\omega)}, \ldots, g^{f(\omega^{d-1})}) \in \mathbb{G}^d$ from $g^{a_0}, \ldots, g^{a_d}$. The detail of the algorithm is described below.

Algorithm 1 Blackbox Fast Fourier Transform

Input : $d = 2^k \in \mathbb{N}$, $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}} \in \mathbb{G}^d$ for $f(x) = a_{d-1}x^{d-1} + \cdots + a_0 \in \mathbb{F}_p[x]$ and the powers $\omega, \omega^2, \ldots, \omega^{d-1}$ of a primitive d-th root of unity $\omega \in \mathbb{F}_p$

Output : $g^{DFT_\omega(f)} := (g^{f(1)}, g^{f(\omega)}, \ldots, g^{f(\omega^{d-1})}) \in \mathbb{G}^d$

1. If $d = 1$ then return g^{a_0}
2. $g^{r_0(x)} \leftarrow (g^{a_0 + a_{d/2}}, \cdots, g^{a_{d/2-1} + a_{d-1}})$,
   $g^{r_1(x)} \leftarrow (g^{1 \cdot (a_0 - a_{d/2})}, \cdots, g^{w^{d/2-1}(a_{d/2-1} - a_{d-1})})$
3. call the algorithm recursively to get $g^{DFT_{w^2}(r_0)}$ and $g^{DFT_{w^2}(r_1)}$
4. return $\left(g^{a(1)}, g^{a(\omega)}, \ldots, g^{a(\omega^{d-1})}\right)$

Lemma 3.3.1. *Given $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ for a polynomial $f(x) = a_{d-1}x^{d-1} + \cdots + a_0$ of degree $< d$, Algorithm 1 runs in $O(d \log d)$ group operations.*

Proof. The correctness of the algorithm follows from the original Fourier transform algorithm. Let $S(d)$ and $T(d)$ denote the number of exponentiations and multiplications in $\mathbb{G}$, respectively, that the algorithm requires for input size d. The cost for the individual steps is: In step 2, d multiplications (divisions) and $d/2$ exponentiations by powers $\omega, \omega^2, \ldots, \omega^{d/2}$ in $\mathbb{G}$, in step 3, $2S(d/2)$ exponentiations and $2T(d/2)$ multiplications. Thus $S(d) = 2S(d/2) + d$, $T(d) = 2T(d/2) + d/2$, and by unfolding the recursions we find that $S(d) = d \log d$ and $T(d) = \frac{1}{2} d \log d$. □

Let $*$ denote the convolution map $f * h = f(x) \cdot h(x) \mod x^d - 1$ for polynomials f and h. Then $DFT_\omega(f * h) = DFT_\omega(f) \cdot DFT_\omega(h)$, where

$\cdot$ denotes the pointwise multiplication. Especially, it is easy to check that $f * h(x) = f(x)h(x)$ when $\deg(f(x)h(x)) < d$. The map $DFT_\omega : f(x) = (f_0, f_1, \ldots, f_{d-1}) \mapsto \big(f(1), f(\omega), \ldots, f(\omega^{d-1})\big)$ can be considered as multiplication by the matrix

$$V_\omega := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{d-1} \\ 1 & \omega^2 & \cdots & \omega^{2(d-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{d-1} & \cdots & \omega^{(d-1)^2} \end{pmatrix}.$$

We can easily verify that $V_\omega \cdot V_{\omega^{-1}} = dI$, where I denotes the $d \times d$ identity matrix. Therefore

$$DFT_\omega^{-1} = V_\omega^{-1} = \frac{1}{n} V_{\omega^{-1}} = \frac{1}{d} DFT_{\omega^{-1}}.$$

With the convolution map, we can obtain the fast multiplication in the blackbox manner.

Lemma 3.3.2. *For given $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ and $h(x) \in \mathbb{F}_p[x]$ with $\deg(f \cdot h) < d$, we can compute $g^{f(x)h(x)}$ in $O(d \log d)$ group exponentiations.*

Proof. The correctness follows from

$$f(x) \cdot h(x) = DFT_\omega^{-1}(DFT_\omega(f * h)).$$

The cost for each step becomes: in step 1, $d - 2$ multiplication by $\omega \in \mathbb{F}_p$ and step 2 $O(d \log d)$ operations in $\mathbb{F}_p$. In step 3 and 5, we require $O(d \log d)$ group exponentiations by Lemma 3.3.1, in step 4 $O(d)$ exponentiations are needed. $\square$

Subsequently, we propose the fast polynomial division algorithm where the coefficients of one of the input polynomial are given in the exponentiated form. As usually, it follows from the Newton iteration method.

Algorithm 2 Fast Convolution in Blackbox Manners

Input : $g^{f(x)}$ and $h(x)$ where $f(x), h(x) \in \mathbb{F}_p[x]$, and a primitive d-th root of unity $\omega \in \mathbb{F}_p$.

Output : $g^{f(x)h(x)} \in \mathbb{G}^d$.

1. compute $\omega^2, \ldots, \omega^{d-1}$
2. $H \leftarrow DFT_\omega(h) = \big(b(1), b(\omega), \ldots, b(\omega^{d-1})\big)$
3. $g^F \leftarrow g^{DFT_\omega(f)} = \Big(g^{a(1)}, g^{a(\omega)}, \ldots, g^{a(\omega^{d-1})}\Big)$
4. $g^C = g^{DFT_\omega(f*h)} \leftarrow (g^F)^H$, pointwise exponentiation
5. return $g^{DFT_\omega^{-1}(C)} = \big(g^{DFT_{\omega^{-1}}(C)}\big)^{\frac{1}{d}}$

We define the reversal of a polynomial $f(x)$ by $\mathrm{rev}_k(f) = x^k f(1/x)$. Observe that if $k = \deg(f)$ then the reversal of f is simply the polynomial with the coefficients of f reversed. We want to find polynomials q and r such that $f = hq + r$ with $\deg(r) < \deg(h)$. By the definition of the reversal we can easily obtain

$$\begin{aligned} \mathrm{rev}_d(f) &= \mathrm{rev}_m(h)\mathrm{rev}_{d-m}(q) + x^{d-m+1}\mathrm{rev}_{m-1}(r) \\ &= \mathrm{rev}_m(h)\mathrm{rev}_{d-m}(q) \mod x^{d-m+1} \end{aligned}$$

where $\deg(f) = d, \deg(h) = m$. Then $\mathrm{rev}_{d-m}(q) = \mathrm{rev}_d(a)\mathrm{rev}_m(h)^{-1} \mod x^{d-m+1}$, and we can find $\mathrm{rev}_m(h)^{-1} \mod x^{d-m+1}$ by using Newton iteration. Consequently we can obtain q and $r = f - hq$.

Lemma 3.3.3. *Given $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ and $h(x)$, we compute $g^{f(x) \mod h(x)}$ in $O(d \log d)$ group exponentiations, where* $\deg(a) = 2d, \deg(b) = d$.

Proof. The cost for individual steps becomes: in step 1, $O(m \log m)$ field operations, and $O((d-m)\log(d-m))$ group exponentiations by lemma 3.3.2

Algorithm 3 Blackbox Polynomial Division

Input : $g^{a(x)}$ and $b(x)$ where $a(x), b(x) \in \mathbb{F}_p[x]$ with $\deg(a) = d, \deg(b) = m$ $(d > m)$

Output : $g^{f(x) \mod h(x)} \in \mathbb{G}^{m-1}$

1. Compute $\mathrm{rev}_m(h)^{-1} \mod x^{d-m+1}$ using Newton iteration.
2. Call the algorithm 2 to compute $g^{q(x)} = g^{\mathrm{rev}_d(f)\mathrm{rev}_m(h)^{-1} \mod x^{d-m+1}}$
3. Call the algorithm 2 to compute $g^{q(x)h(x)}$ with inputs $g^{q(x)}$ and $h(x)$
4. Return $g^{r(x)} = g^{f(x)}/g^{q(x)h(x)}$

---

in step 2, and $O(d \log d)$ group exponentiations in step 3. Finally in step 4 we only require d divisions of the group elements. Especially if $\deg(f) = 2d, \deg(h) = d$, then the total cost becomes $O(d \log d)$ group exponentiations. □

Finally we can propose the fast multipoint evaluation algorithm by building up the sub-product tree of $(x-r_0)(x-r_1)\cdots(x-r_{d-1})$ where $r_0, \ldots, r_{d-1}$ are values to be evaluated. Let $m_i := x - r_i$ and define the recursive relations $M_{0,j} = m_j$, $M_{i+1,j} = M_{i,2j} \cdot M_{i,2j+1}$. From the fact that $f(r_j) = f(x) \mod m_j$, we can obtain the following algorithm and the lemma is just a direct consequence. We will omit the proof of the lemma.

Lemma 3.3.4. *Given $g^{f(x)} = (g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}})$, we can compute $g^{f(r_0)}, \cdots, g^{f(r_{d-1})}$ in $O(d \log^2 d)$ exponentiations in group $\mathbb{G}$.*

Until now, we assumed that existence of a d-th primitive root of unity in $\mathbb{F}_p$, i.e. $d|(p-1)$ for the fast multipoint evaluation in the blackbox manner. However, to apply this method to our algorithm solving the DLPwAI, the divisibility of d should be unrestricted.

Algorithm 4 Blackbox Multipoint Evaluation Algorithm

Input : $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ with $f(x) \in \mathbb{F}_p[x]$ of degree $< d = 2^k$ for some $k \in \mathbb{N}$ and $r_0, \ldots, r_{d-1} \in \mathbb{F}_p$

Output : $g^{f(r_0)}, \ldots, g^{f(r_{n-1})} \in \mathbb{G}$

1. Compute the subproduct $M_{i,j}$
2. Call the algorithm 3, $g^{R_0} \leftarrow g^{f(x) \mod M_{k-1,0}}$, $g^{R_1} \leftarrow g^{f(x) \mod M_{k-1,1}}$
3. Call the algorithm recursively to compute $g^{R_0(x_0)}, \ldots, g^{R_0(x_{d/2-1})}$
4. Call the algorithm recursively to compute $g^{R_1(x_{d/2})}, \ldots, g^{R_1(x_{d-1})}$
5. Return $g^{R_0(x_0)}, \ldots, g^{R_0(x_{d/2-1})}, g^{R_1(x_{d/2})}, \ldots, g^{R_1(x_{d-1})}$

The Schönhage-Straßen multiplication method does not require the existence of the d-th primitive root of unity in $\mathbb{F}_p$. The similar result in the blackbox manner can be easily obtained.

Let $f(x) = a_{d-1}x^{d-1} + \cdots + a_0$ and $h(x) = b_{d-1}x^{d-1} + \cdots + b_0$ be polynomials over $\mathbb{F}_p$ with $\deg(f \cdot h) < d = 2^k$. The blackbox Schönhage-Straßen multiplication outputs $g^{f(x)h(x)} = (g^{c_0}, g^{c_1}, \ldots, g^{c_{d-1}})$ with inputs $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ and $h(x) = b_0 + \cdots + b_{d-1}x^{d-1}$.

Let us first explain the non-blackbox version of the method. Let $m = 2^{\lfloor k/2 \rfloor}$ and $t = d/m$. Write down the polynomial as $f(x) = A_0 + A_1x^m + \cdots + A_{t-1}x^{m(t-1)}$ where $A_i \in \mathbb{F}_p[x]$ with degree less than m and let $f'(x, y) = A_0 + A_1y + \cdots + A_{t-1}y^{t-1} \in \mathbb{F}_p[x, y]$ so that $f'(x, x^m) = f(x)$. Consider a ring $D := \mathbb{F}_p[x]/(x^{2m} + 1)$ and let $\zeta \in D$ be an element corresponding to x in $\mathbb{F}_p[x]/(x^{2m} + 1)$. Then we can view $f^*(y) = a'(\zeta, y) = A_0(\zeta) + A_1(\zeta)y + \cdots + A_{t-1}(\zeta)y^{t-1} \in D[y]$. The goal is to obtain $f(x)h(x) \mod x^d + 1$ which is equivalent to $f^*(y)h^*(y) \mod y^t + 1$. However since $\zeta^{2m} = -1$ and

$\zeta^{4m} = 1$, ζ is a $4m$-th primitive root of unity in D, thus $\eta = \zeta^2$ if $t = m$ and $\eta = \zeta$ if $t = 2m$ is a primitive $2t$-th root of unity in D. Now we want to compute $f^*(\eta y)h^*(\eta y) \mod (\eta y)^t + 1$ or $f^*(\eta y)h^*(\eta y) \mod y^t - 1$, this can be done by fast multiplication using the discrete Fourier transform with the t-th primitive root of unity $\omega = \eta^2$ in D. The multiplication in D can be done recursively with polynomial degree less than $2m$. In blackbox version of the algorithm, we simply write $g^{f(x)} = (g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}) = (g^{A_0}, \ldots, g^{A_{t-1}})$ where g^{A_i} means $(g^{a_{mi}}, g^{a_{mi+1}}, \ldots, g^{a_{mi+(m-1)}})$.

Finally, we give the blackbox version of fast blackbox Schönhage and Straßen multiplication in the following algorithm.

Algorithm 5 Blackbox Schönhgae-Straßen Multiplication

Input : $d = 2^k \in \mathbb{N}$, $g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}}$ and $h(x) = b_{d-1}x^{d-1} + \cdots + b_0$ where $f(x), h(x) \in \mathbb{F}_p[x]$ with $\deg(fh) < d$

Output : $g^{f(x)h(x)} := (g^{c_0}, g^{c_1}, \ldots, g^{c_{d-1}}) \in \mathbb{G}^d$

1. $m \leftarrow 2^{\lfloor k/2 \rfloor}$, $t \leftarrow d/m$
   let $g^{f(x)} = (g^{A_0}, \ldots, g^{A_{t-1}})$ and $h(x) = (B_0, \ldots, B_{t-1})$ so that $f(x) = \sum_{i=0}^{t-1} A_i x^{mi}$, $h(x) = \sum_{i=0}^{t-1} B_i x^{mi}$ where $\deg A_i, \deg B_j < m$

2. let $D = \mathbb{F}_p[x]/(x^{2m}+1)$ and $\zeta \leftarrow x \mod (x^{2m}+1)$
   if $t = 2m$ then $\eta \leftarrow \zeta$, otherwise $\eta \leftarrow \zeta^2$ so that η is a primitive $2t$-th root of unity
   call the algorithm 2 with $\omega = \eta^2$ to compute $g^{c^*(\eta y)} = g^{f^*(\eta y)h^*(\eta y) \mod (y^t-1)}$ using algorithm 5 recursively for the multiplication in D

3. return $g^{c^*(y)} = (g^{C_0}, \ldots, g^{C_{t-1}})$

Lemma 3.3.5. *Let $f(x), h(x) \in \mathbb{F}_p[x]$ with $\deg(fh) < d$. Given $g^{f(x)}$ and*

$h(x)$, we can compute $g^{f(x)h(x)}$ in $O(d \log d \log\log d)$ group operations.

3.4 Balls-and-Bins Problem

In this section, we shall briefly review and discuss further on the birthday problem which is generally called the balls-and-bins problem. The balls-and-bins problem considers the followings: there exist balls and N bins, and we pick up a ball and put into a bin (the ball is put into each bin with certain probability), and iterate this process until two different balls are put into one bin, which we call a collision. Then the problem asks the expected number of the trials until the collision occurs. Typically, the birthday problem refers to the balls-and-bins problem when a ball is put into N bins equiprobably. There also have been many researches considering the balls with the several types and finding a collision between two different types of balls when the probability is not uniform [23, 43, 50].

Throughout the paper, we assume that the probability only depends on the bins, not on the ball.

3.4.1 Balls-and-Bins Problem with Uniform Probability

Suppose that each balls are put into N bins, numbered by $1, \ldots, N$, with the equiprobability. Denote p_i by the probability that a ball is put into the bin numbered i. We write the vector of the probability as $(p_1, \ldots, p_N) = (\frac{1}{N}, \ldots, \frac{1}{N})$. In this case, the problem is the classical birthday problem.

Let Z be a random variable that indicates the number of the trials until the first collision occurs. The probability that no collision occurs until r

trials, $P[Z \leq r]$, is given by

$$\frac{N}{N} \cdot \frac{N-1}{N} \cdot \frac{N-2}{N} \cdots \frac{N-(r-1)}{N} = 1 - \frac{N-1}{N} \cdot \frac{N-2}{N} \cdots \frac{N-(r-1)}{N}.$$

From $e^x \geq 1 + x$, we have $e^{-j/N} \geq 1 - \frac{j}{N}$ and

$$1 - \frac{N-1}{N} \cdot \frac{N-2}{N} \cdots \frac{N-(r-1)}{N} \geq 1 - e^{-1/N} \cdots e^{-(r-1)/N} = 1 - e^{-\frac{(r-1)r}{2N}}.$$

The last term is approximately close to $1 - 1/e = 0.632\cdots$, when $r^2 \approx N$. On the other hand, the expected number of the trials is $E[Z] \approx \sqrt{\frac{\pi N}{2}}$ for $N \to \infty$.

3.4.2 Balls-and-Bins Problem with Non-Uniform Probability

In this section, we consider the non-uniform balls-and-bins problem, where the probabilities $p_1, \ldots, p_N$ are not equiprobable. Suppose that we have N bins numbered from 1 to N and for $i = 1, 2, \ldots, N$, define p_i by the probability that a randomly chosen ball is put into a bin of the number i. We say that a collision occurred when at least one bin contains at least two balls in it. We say that throwing a ball into a bit as one trial. This section is contributed by J. H. Seo.

Let S_r be the probability that a collision occurs in r trials. In this section, we shall show that S_r is non-negligible for $r \approx \sqrt{1/\sum_i p_i^2}$. Define $E_i^{(r)}$ by an event that a collision occurs in a bin of a number i after r trials. Then we have

$$\begin{aligned} S_r &= \Pr(E_1^{(r)} \cup \cdots \cup E_N^{(r)}) = \sum_{k=1}^{N} (-1)^{k+1} \sum_{1 \leq i_1 \neq i_2 \neq \cdots \neq i_k \leq N} \Pr(E_{i_1}^{(r)} \cap \cdots \cap E_{i_k}^{(r)}) \\ &\geq \sum_{i=1}^{N} \Pr(E_i^{(r)}) - \sum_{1 \leq i \neq j \leq N} \Pr(E_i^{(r)} \cap E_j^{(r)}). \end{aligned}$$

Unless there is no ambiguity, we shall omit the superscript (r) in $E_i^{(r)}$.

Definition 3.4.1. Consider the r-tuple $\vec{b} := (b_1, \ldots, b_r) \in [1, N]^r$, where $[1, N]$ is a set of integers from 1 to N. For $k = 1, \ldots, N$, define $\mathbf{wt}_{(k)}(\vec{b})$ by the size of a set $\{1 \leq i \leq r : b_i = k\}$. Let $B_{r,k}^{(i)} := \{\vec{b} = (b_1, \ldots, b_r) : \mathbf{wt}_{(k)}(\vec{b}) = i\}$.

Proposition 3.4.1. *With the notations as above, we have*

$$\begin{aligned}
\Pr(E_k) &= \sum_{i \geq 2} \sum_{\vec{b} \in B_{r,k}^{(i)}} p_{b_1} \cdots p_{b_r} = 1 - \left(\sum_{\vec{b} \in B_{r,k}^{(1)}} p_{b_1} \cdots p_{b_r} + \sum_{\vec{b} \in B_{r,k}^{(0)}} p_{b_1} \cdots p_{b_r} \right) \\
&= 1 - \big(r \cdot p_k \cdot (1 - p_k)^{r-1} + (1 - p_k)^r \big) \\
&= 1 - (1 - p_k)^{r-1} \cdot (1 + (r-1)p_k).
\end{aligned}$$

Proof. The summation $\sum_{\vec{b} \in B_{r,k}^{(1)}} p_{b_1} \cdots p_{b_r}$ means that only one ball is put into a bin k until r trials, and the other summation taken over $B_{r,k}^{(0)}$ means the probability that no ball is thrown to any bin. Thus the results are easily verified. □

Lemma 3.4.1. *Let* $S := 1 + 2(1-x) + 3(1-x)^2 + \cdots + (r-1) \cdot (1-x)^{r-2}$, *then we have*

$$(1-x)^{r-1} \cdot (1 + (r-1)x) = 1 - x^2 \cdot S.$$

Proof. It follows from

$$\begin{aligned}
S - (1-x)S &= 1 + (1-x) + \cdots + (1-x)^{r-2} - (r-1)(1-x)^{r-1} \\
&= \frac{(1-x)^{r-1} - 1}{(1-x) - 1} - (r-1)(1-x)^{r-1} \\
&= \frac{(1-x)^{r-1} \cdot (1 + (r-1)x) - 1}{(-x)}.
\end{aligned}$$

□

From Proposition 3.4.1 and Lemma 3.4.1, the following is easily deduced,

$$\begin{aligned}\Pr(E_k) &= p_k^2 \cdot [1 + 2\cdot(1-p_k) + \cdots + (r-1)\cdot(1-p_k)^{r-2}] \\ &\geq p_k^2 \cdot [1 + 2\cdot(1-p_k) + \cdots + (r-1)\cdot(1-(r-2)p_k)] \\ &\geq p_k^2 \cdot \left[\frac{(r-1)r}{2}\cdot(1-(r-2)p_k)\right].\end{aligned}$$

Now let us consider the upper bound of $\Pr(E_k \cap E_\ell)$.

Proposition 3.4.2. *With the notations as above, we have*

$$\begin{aligned}\Pr(E_k \cap E_\ell) &= \sum_{i,j\geq 2} \sum_{\vec{b}\in B_{r,k}^{(i)}\cap B_{r,\ell}^{(j)}} p_{b_1}\cdots p_{b_r} \leq \binom{r}{2}\cdot\binom{r-2}{2}\cdot p_k^2\cdot p_\ell^2 \\ &= \frac{r(r-1)(r-2)(r-3)}{4} p_k^2 \cdot p_\ell^2.\end{aligned}$$

Proof. For any $i \geq 2$ and $j \geq 2$, $\vec{b} \in B_{r,k}^{(i)} \cap B_{r,\ell}^{(j)}$ is of form $(b_1, b_2, \ldots, b_r)$ for $b_i = b_j = k$ and $b_s = b_t = \ell$ with $i \neq j$ and $s \neq t$. And in that case, we have $p_{b_1}\cdots p_{b_r} \leq p_k^2 \cdot p_\ell^2$. The value $\binom{r}{2}$ indicates the possible number of two positions for k and $\binom{r-2}{2}$ stands for the possible number of the other two positions of ℓ. $\square$

From the above results, for $r < 1/(2\cdot\max_k\{p_k\})$, we have the following inequality

$$\begin{aligned}S_r &\geq \sum_{k=1}^{N}\Pr(E_k) - \sum_{1\leq k\neq\ell\leq N}\Pr(E_k\cap E_\ell) \\ &\geq \frac{(r-1)r}{4}\cdot\sum_{1\leq k\leq N} p_k^2 - \frac{r^2(r-1)^2}{4}\cdot\sum_{1\leq k\neq\ell\leq N} p_k^2 p_\ell^2 \\ &= \frac{(r-1)r}{4}\cdot\sum_{1\leq k\leq N} p_k^2 - \frac{r^2(r-1)^2}{4}\cdot\left\{\left(\sum_{1\leq k\leq N} p_k^2\right)^2 - \left(\sum_{1\leq k\leq N} p_k^4\right)\right\}.\end{aligned}$$

The last term in the above inequality is maximized by $1/16+\epsilon$ for $\epsilon = \frac{r^2(r-1)^2}{4}\cdot\sum_k p_k^4$, when $(r-1)r\cdot\sum_k p_k^2 = 1/2$. Thus we expect a collision with non-negligible probability after $r \approx \sqrt{\frac{1}{2(\sum_k p_k^2)}}$.

3.5 Polynomials with small value sets

In this section, we introduce a new approach to solve the DLPwAI using the polynomials with the small value sets.

We briefly describe the idea: first, we compute two lists $\{g^{f(r_1\alpha)}, \dots, g^{f(r_m\alpha)}\}$ and $\{g^{f(s_1)}, \dots, g^{f(s_m)}\}$ for given $g, g^{\alpha}, \dots, g^{\alpha^d}$ and random $r_i, s_j \in \mathbb{F}_p$. If there exists a collision between two lists, say $g^{f(r_i\alpha)} = g^{f(s_j)}$, then we solve the equation $f(r_i\alpha) = f(s_j)$ in the intermediate α. Since the degree of $f(x)$ is d, we obtain at most d candidates for α. Finally, we can find a solution α by d times of exhaustive search. Hence, the important parts of the algorithm are to obtain a polynomial such that the expected number of m until collision occurs is small and to compute the list $g^{f(r_1\alpha)}, \dots, g^{f(r_m\alpha)}$ efficiently for given $g, g^{\alpha}, \dots, g^{\alpha^d}$.

3.5.1 An approach using the polynomial of small value set: uniform case

In this section, we observe how to reduce the DLPwAI into finding a polynomial with the small value set.

Define the value set of a polynomial $f(x) \in \mathbb{F}_p[x]$ by $V(f) := \{f(x) : x \in \mathbb{F}_p\} = \{a_1, \dots, a_v\}$, where t is the size of value set. We consider the balls-and-bins problem with respect to the polynomial map by f. For randomly chosen element r (a random ball) from $\mathbb{F}_p$, we assume that the ball r is put in a bin numbered by $f(r)$. The collision means that there exists different elements r and s such that $f(r) = f(s)$.

Denote p_a by the probability that a random ball $r \in \mathbb{F}_p$ takes the value $f(r) = a$ for $a \in V(f)$. In this section, for a while, we assume that $p_{a_1} = \cdots = p_{a_v}$ for all $a_i \in V(f)$ so that the probability vector is given by $(p_{a_1}, \dots, p_{a_v}) =$

$(\frac{1}{v}, \ldots, \frac{1}{v})$. Since the expected number of trials until the collision is $O(\sqrt{v})$, after taking the value for $m := O(\sqrt{v})$ elements, we expect the collision $f(r) = f(s)$. Consider two lists for random r_i and s_j for $1 \leq i, j \leq m$:

$$L_1 := \{g^{f(r_1\alpha)}, \ldots, g^{f(r_m\alpha)}\}$$

and

$$L_2 := \{g^{f(s_1)}, \ldots, g^{f(s_m)}\}.$$

By the birthday problem, there exists a collision with high probability. We have a collision, say $f(r_i\alpha) = f(s_j)$. It gives an equation of degree d in the intermediate α. We can solve the equation by finding the roots of a polynomial $\tilde{f}(x) := f(r_i x) - f(s_j)$ of degree d which costs the expected number of $O(d \cdot \log q \cdot (\log d)^2 \cdot \log\log d)$ operations in $\mathbb{F}_p$ [56] (we assume that the multiplication in $\mathbb{F}_p$ is done by Schönhage-Straßen method). Using the fast multipoint evaluation method described in the previous section, the list L_1 can be computed in $O(m \log d \log\log d)$ group operations. Computing the list L_2 costs $O(m \log d \log\log d)$ operations in $\mathbb{F}_p$ and $O(m)$ group exponentiations. Thus we have the followings.

Theorem 3.5.1. *Let $g, g^\alpha, \cdots, g^{\alpha^d}$ be given and let $f(x) := f_0 + f_1 x + \cdots + f_d x^d$ be a polynomial over $\mathbb{F}_p$ of degree d. Assume that the preimage set $f^{-1}(a)$ for each $a \in V(f)$ is equally distributed. Then we can solve α in $O((\sqrt{v}\log d + d\log p(\log d)^2) \cdot \log\log d)$ operations (including the field operations and the group operations).*

For fixed d and p, the complexity of Theorem 3.5.1 reduces when v becomes smaller. Thus, Theorem 3.5.1 says that the DLPwAI reduces to find polynomials with small value sets. Finding such polynomials has been old research topics in number theoretic area.

Polynomial with Minimal Value Set

Let $f(x) \in \mathbb{F}_p[x]$ be a polynomial of degree d. For each $a \in V(f)$, the preimage $f^{-1}(a) := \{x : f(x) = a\}$ has at most d elements. If v is the size of $V(f)$, then

$$p = |f^{-1}(a_1)| + \cdots + |f^{-1}(a_v)| \leq d \cdot v,$$

in other words v is an integer satisfying $v \geq \frac{p}{d}$, or equivalently $v \geq \lfloor \frac{p-1}{d} \rfloor + 1$. The polynomial satisfying $v = \lfloor \frac{p-1}{d} \rfloor + 1$ is said to have the minimal value set.

Consider a polynomial $f(x) = x^d$ with $d|(p-1)$. From the divisibility of $p-1$ by d, we have a primitive d-th root of unity ζ_d in $\mathbb{F}_p$ and thus we have $f(\zeta_d x) = f(\zeta_d^2 x) = \cdots = f(\zeta_d^d x)$ for any nonzero x. In other words, $x \mapsto f(x)$ defines a d-to-1 mapping except at $x = 0$. Then the size of $V(f)$ is $v = \frac{p-1}{d} + 1$. Thus $f(x) = x^d$ has the minimal value set for $d|(p-1)$.

Applying Theorem 3.5.1 with this polynomial, we can solve the DLPwAI in $\tilde{O}\left(\sqrt{\frac{p-1}{d}} + d\right)$ which minimizes to $\tilde{O}(p^{1/3})$ at $d \approx p^{1/3}$.

In [9], it is shown that polynomials of form $(x+b)^d + c$ for $b, c \in \mathbb{F}_p$ has the minimal value set when $d|(p-1)$ and these are the only polynomials with minimal value set. With these polynomials, we can solve the DLPwAI in $\tilde{O}(p^{1/3})$ operations.

Polynomials with small value set

On the other hand, it is not an easy task to find a polynomial with small value sets. In [53], Uchiyama showed that $v := |V(f)| = c_d p + O(1)$ on the average, where the average is taken over the monic polynomials of degree d. Here, $c_d = 1 - \frac{1}{2!} + \frac{1}{3!} - \cdots + (-1)^{d-1}\frac{1}{d!} \approx \frac{1}{e}$.

The classification of the polynomials with the value set of size less than $2p/d$ for $d < p^{1/4}$ was given in [26].

3.5.2 Approach using polynomials with almost small value set: non-uniform case

Let $f(x)$ be a polynomial of degree d with the value set $V(f) = \{a_1, \ldots, a_v\}$. Define a set $S_i := \{a \in V(f) : |f^{-1}(a)| = i\}$, $R_i = |S_i|$ and $R = |\{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : f(x) = f(y)\}|$, then we have the following equations,

$$p = \sum_{i=1}^{d} iR_i,\ |V(f)| = \sum_{i=1}^{d} R_i,\ \text{and } R = \sum_{i=1}^{d} i^2 R_i.$$

Now we want to determine the value of m for two lists $\{f(r_1), \ldots, f(r_m))\}$ and $\{f(s_1), \ldots, f(s_m)\}$ have a non-empty intersection for random r_i and s_j.

We consider this problem as the non-uniform balls-and-bins problem with the probability vector,

$$(p_{a_1}, \ldots, p_{a_t}) = (\underbrace{\frac{1}{p}, \ldots, \frac{1}{p}}_{R_1}, \underbrace{\frac{2}{p}, \ldots, \frac{2}{p}}_{R_2}, \cdots, \underbrace{\frac{d}{p}, \ldots, \frac{d}{p}}_{R_d}).$$

From the analysis in Section 3.4.2, we expect a collision within $r \approx \sqrt{1/2 \sum_k p_k^2}$. Note that $\sum_k p_k^2 = \frac{R}{p^2}$. We can restate Theorem 3.5.1.

Theorem 3.5.2. *Let $g, g^{\alpha}, \cdots, g^{\alpha^d}$ be given and let $f(x) := f_0 + f_1 x + \cdots + f_d x^d$ be a polynomial over $\mathbb{F}_p$ of degree d. Let R be defined previously. Then we can solve α in $O((m \log d + d \log p (\log d)^2) \cdot \log\log d)$ operations (including the field operations and the group operations) for $m := O(\sqrt{p^2/R})$.*

The value of R is closely related to the number of the absolutely irreducible factors of $f^*(x, y) = f(x) - f(y)$. The Weil's bound implies that $R = rp + O(d^2\sqrt{p})$, where r is the number of the absolutely irreducible factors of $f^*(x, y)$. Since it is obvious that $1 \le r \le d$, to reduce the complexity it is desirable to find $f(x)$ such that r is close to d. The typical example satisfying $r = d$ is the polynomial $f(x) = x^d$ for $d|(p-1)$ which was given as an example in the previous section.

The other example is given by the Dickson polynomial. For $a \in {\mathbb{F}_p}^*$ and $d \in \mathbb{Z}_{\geq 1}$, the Disckson polynomial is defined by

$$D_d(x,a) = \sum_{k=0}^{\lfloor d/2 \rfloor} \frac{d}{d-k} \binom{d-k}{k} (-a)^k x^{d-2k}.$$

The substitution polynomial $D_d(x,a) - D_d(y,a)$ factorizes into

$$(x-y) \prod_{k=1}^{(d-1)/2} \left(x^2 - (\zeta^k + \zeta^{-k})xy + y^2 + a(\zeta^{2k} + \zeta^{-2k} - 2)) \right)$$

for nonzero $a \in \mathbb{F}_p$, where $\zeta \in \mathbb{F}_{p^2}$ is a primitive d-th root of unity for $d|(p^2-1)$. Thus it satisfies $r = \frac{d}{2}$, and our theorem solves the DLPwAI in $\tilde{O}(\sqrt{\frac{p}{d}} + d)$ with the Dickson polynomial.

On the other hand, it is known that $R_1 = \frac{p-1}{2}$, $R_d = \frac{p+1}{2d}$ and $R_i = 0$ otherwise, when d is a divisor of $p+1$ and a is a quadratic non-residue [15] (the similar result is also verified for a quadratic residue a). Consider the set $V(f) = \{v_1, \ldots, v_\ell\}$ where $f(x) = D_d(x,a)$, then $|V(f)| = \frac{p-1}{2} + \frac{p+1}{2d}$. This means that, roughly speaking, half of the elements in $\mathbb{F}_p$ maps d-to-1 by the polynomial $f(x)$, i.e. the expected number of collision is $O(\sqrt{p/2d})$ assuming all the elements were chosen from that half of the domain.

3.5.3 Generalization of the Dickson Polynomial and its value set

In this section, we consider the value set of the generalized Dickson polynomial of degree d in two variable. For a fixed $a \in \mathbb{F}_p$, consider the polynomial

$$f_a(z) = z^3 - xz^2 + yz - a = (z-\sigma_0)(z-\sigma_1)(z-\sigma_2).$$

The generalized Dickson polynomial is given by

$$D_d^{(1)}(x,y,a) := \sigma_0^d + \sigma_1^d + \sigma_2^d,$$

and

$$D_d^{(2)}(x, y, a) := \sigma_0^d \sigma_1^d + \sigma_1^d \sigma_2^d + \sigma_2^d \sigma_1^d.$$

Generally, the Dickson polynomial of degree in n variables is defined by

$$D_d^{(i)}(x_1, x_2, \ldots, x_n, a) = S_i(\sigma_0^d, \sigma_1^d, \ldots, \sigma_n^d),$$

where σ_i's are roots of the polynomial

$$f_a(z) = z^{n+1} - x_1 z^n + \cdots + (-1)^n x_n z + (-1)^{n+1} a$$

and the polynomial S_i is the i-th symmetric polynomial in $(n+1)$ variables. The one variable case coincides with the original Dickson polynomial. The value sets of the Dickson polynomial $D_d(x, a)$ was given in [15].

We try to count the value sets of the $(D_d^{(1)}(x, y, a), D_d^{(2)}(x, y, a)) \in \mathbb{F}_p \times \mathbb{F}_p$. Unless there is an ambiguity, we simply write $D_d^{(1)} = D_1$ and $D_d^{(2)} = D_2$. Consider the partition

$$\begin{aligned} \mathbb{F}_p \times \mathbb{F}_p &= \{(x, y) : z^3 - xz^2 + yz - a \text{ is irreducible over } \mathbb{F}_p\} \\ &\quad \cup \{(x, y) : z^3 - xz^2 + yz - a \text{ is reducible over } \mathbb{F}_p\}. \end{aligned}$$

Define the former set by $\mathrm{Irr}(a)$ and the later by $\mathrm{Red}(a)$.

From now on, assume that

$$d | \Phi_3(p) = p^2 + p + 1$$

so that the primitive d-th root of unity $\zeta := \zeta_d$ exists in $\mathbb{F}_{p^3} \backslash \mathbb{F}_p$.

Lemma 3.5.1. *The pair $(x, y) \in Irr(a)$ if and only if there exists $\sigma \in \mathbb{F}_{p^3} \backslash \mathbb{F}_p$ such that $z^3 - xz^2 + yz - a = (z - \sigma)(z - \sigma^p)(z - \sigma^{p^2})$.*

Proof. Obvious from the definition, since σ, σ^p and σ^{p^2} are the conjugates. $\square$

Lemma 3.5.2. *Consider the pairs* $(x_0, y_0), (x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ *such that*

$$f_{a,0} := z^3 - x_0 z^2 + y_0 z - a = (z - \sigma_0)(z - \sigma_1)(z - \sigma_2)$$

and

$$f_a := z^3 - xz^2 + yz - a = (z - \tau_0)(z - \tau_1)(z - \tau_2),$$

then $D_1(x, y) = D_1(x_0, y_0)$ *and* $D_2(x, y) = D_2(x_0, y_0)$ *if and only if* $\{\sigma_0^d, \sigma_1^d, \sigma_2^d\} = \{\tau_0^d, \tau_1^d, \tau_2^d\}$ *(the set equality).*

Proof. Note that

$$z^3 - D_1(x, y)z^2 + D_2(x, y)z - a^d = (z - \tau_0^d)(z - \tau_1^d)(z - \tau_2^d)$$

and

$$z^3 - D_1(x_0, y_0)z^2 + D_2(x_0, y_0)z - a^d = (z - \sigma_0^d)(z - \sigma_1^d)(z - \sigma_2^d)$$

by the definition of the Dickson polynomial. Since these two polynomials are the same, the roots are the same. (Note that $f_{z,0}$ and f_z may be reducible over $\mathbb{F}_p$.) □

Lemma 3.5.3. *Fix* $(x_0, y_0) \in Irr(a)$ *with the corresponding root* $\sigma \in \mathbb{F}_{p^3}$*, if* $(x, y) \in Irr(a)$ *such that* $D_1(x, y) = D_1(x_0, y_0)$ *and* $D_2(x, y) = D_2(x_0, y_0)$*, then*

$$x = x_i := (\sigma\zeta^i) + (\sigma\zeta^i)^p + (\sigma\zeta^i)^{p^2}$$

and

$$y = y_i := (\sigma\zeta^i) \cdot (\sigma\zeta^i)^p + (\sigma\zeta^i)^p \cdot (\sigma\zeta^i)^{p^2} + (\sigma\zeta^i)^{p^2} \cdot (\sigma\zeta^i),$$

for the primitive d*-th root of unity* ζ *and* $i = 0, 1, \ldots, d-1$.

Proof. Since $(x_0, y_0) \in \mathrm{Irr}(a)$, we write $z^3 - x_0z^2 + y_0z - a = (z - \sigma)(z - \sigma^p)(z - \sigma^{p^2})$ for some $\sigma \in \mathbb{F}_{p^3}$, and also similarly for $(x, y) \in \mathrm{Irr}(a)$ with $\tau \in \mathbb{F}_{p^3}$.

From Lemma 3.5.2, we have $\tau^d = \sigma^d$ or $\tau^d = (\sigma^p)^d$ or $\tau^d = (\sigma^{p^2})^d$. In the first case, $\tau = \sigma \cdot \zeta^i$ for the primitive d-th root of unity and $i = 0, 1, \ldots, d-1$ and $x = (\sigma\zeta^i) + (\sigma\zeta^i)^p + (\sigma\zeta^i)^{p^2}$. In the second case, we have $\tau = \sigma^p \cdot \zeta^i$. Since $\gcd(d, p) = 1$, the value ζ^p is another primitive d-th root of unity, thus we can write $\tau = \sigma^p \cdot (\zeta^p)^j$ for some j such that $\zeta^i = \zeta^{pj}$. In this case, $x = \tau + \tau^p + \tau^{p^2} = (\sigma\zeta^j)^p + (\sigma\zeta^j)^{p^2} + (\sigma\zeta^j)$ and similarly for y. So, the counting is duplicated. We also have the similar result for the third case. $\square$

Lemma 3.5.4. *Let (x, y) and (x_0, y_0) be as described in Lemma 3.5.3. Let γ be a primitive element of $\mathbb{F}_{p^3}$. Then $x_i = x_0$ and $y_i = y_0$ for some $0 < i < d$ if and only if $\sigma \in M := \langle \gamma^{\frac{p^2+p+1}{d}} \rangle \cap (\mathbb{F}_{p^3} \backslash \mathbb{F}_p)$.*

Proof. By the same argument in Lemma 3.5.2, $(x_i, y_i) = (x_0, y_0)$ if and only if $\{\sigma, \sigma^p, \sigma^{p^2}\} = \{\sigma\zeta^i, (\sigma\zeta^i)^p, (\sigma\zeta^i)^{p^2}\}$. If $\sigma = \sigma\zeta^i$, it leads only trivial case $\zeta^i = 1$, i.e. d divides i. And $\sigma^p = \sigma\zeta^i$ if and only if $\sigma^{p-1} = \zeta^i$, thus $\sigma = \gamma^{\frac{p^2+p+1}{d} \cdot i}$. Since $d | (p^2 + p + 1)$ and $\gcd(p + 1, p^2 + p + 1) = 1$ for prime p, the third case also deduces that $\sigma = \gamma^{\frac{p^2+p+1}{d} \cdot i}$. $\square$

Definition 3.5.1. For fixed $(x_0, y_0) \in \mathrm{Irr}(a)$, define the set of elements $(x, y) \in \mathrm{Irr}(a)$ such that $(D_1(x, y), D_2(x, y)) = (D_1(x_0, y_0), D_2(x_0, y_0))$ by

$$I(x_0, y_0) := |\{(x, y) \in \mathrm{Irr}(a) : (D_1(x, y), D_2(x, y)) = (D_1(x_0, y_0), D_2(x_0, y_0))\}|.$$

Theorem 3.5.3. *For fixed $(x_0, y_0) \in Irr(a)$, we have $I(x_0, y_0) = d$ if and only if $z^3 - x_0 z^2 + y_0 z - a = (z - \sigma)(z - \sigma^p)(z - \sigma^{p^2})$ for $\sigma \notin M$.*

Proof. From Lemma 3.5.3, $I(x_0, y_0) \leq d$. If $\sigma \in M$, then by Lemma 3.5.4, there exists some $0 < i < d$ satisfying $x_i = x_0$ and $y_i = y_0$. So, we have $I(x_0, y_0) < d$. Conversely, if $I(x_0, y_0) < d$, Lemma 3.5.4 asserts that $x_i = x_0$ and $y_i = y_0$ for some $0 < i < d$ yielding σ must be in M. $\square$

Definition 3.5.2. Define the set of the irreducible polynomials $f_\sigma(z) = (z-\sigma)(z-\sigma^p)(z-\sigma^{p^2})$ of degree 3 with $\sigma \in M$ and $\sigma^{1+p+p^2} = a$ by $\mathrm{Irr}_M(a)$. In other words, $\mathrm{Irr}_M(a) := \{f_\sigma(z) = (z-\sigma)(z-\sigma^p)(z-\sigma^{p^2}) \in \mathrm{Irr}(a) : \sigma \in M\}$. For $a \in \mathbb{F}_p$, we denote $\iota(a) := |\mathrm{Irr}(a)|$ and $\iota_M(a) := |\mathrm{Irr}_M(a)|$.

The following is a direct consequence of Theorem 3.5.3.

Corollary 3.5.1. *For $a \in \mathbb{F}_p$ and the Dickson polynomial (D_1, D_2) in two variables of degree d, we have $\{(x,y) \in Irr(a) : I(x,y) = d\} = Irr(a) \backslash Irr_M(a)$.*

By Corollary 3.5.1, $\iota(a) - \iota_M(a)$ describes the size of the preimage in $\mathrm{Irr}(a) \subseteq \mathbb{F}_p \times \mathbb{F}_p$ of the two variable Dickson polynomial which maps d to 1. Since the ratio of the irreducible polynomials over the polynomials of degree d is approximately $\frac{1}{d}$, thus we have $\iota(a) \approx \frac{p^2}{3}$. The following lemma shows that $\iota_M(a)$ is relatively small compared to $\iota(a)$ when $d \ll p^2$.

Lemma 3.5.5. *For fixed $a \in \mathbb{F}_p$, $\iota_M(a) = d$ or $3d$.*

Proof. Let γ be a primitive element of $\mathbb{F}_{p^3}$. Since $a \in \mathbb{F}_p$, we write $a = \gamma^{(p^2+p+1)\cdot k}$ for some $0 \le k \le p-1$. On the other hand, $\sigma = \gamma^{\frac{p^2+p+1}{d}\cdot i}$ for some $0 \le i \le d(p-1)$ since $\sigma \in M$. And $\sigma^{1+p+p^2} = \left(\gamma^{\frac{p^2+p+1}{d}\cdot i}\right)^{1+p+p^2} = a = \gamma^{(p^2+p+1)\cdot k}$. Thus $\iota_M(a)$ is the number of $0 \le i \le d(p-1)$ satisfying

$$\frac{p^2+p+1}{d} \cdot i \equiv k \pmod{p-1},$$

or, equivalently

$$(p^2+p+1) \cdot i \equiv dk \pmod{d(p-1)}.$$

It follows $\iota_M(a) = \gcd(p^2+p+1, d(p-1)) = d \cdot \gcd(\frac{p^2+p+1}{d}, p-1) = d$ or $3d$. $\square$

From the above, we deduce that most of the elements in $\mathrm{Irr}(a)$ maps d-to-1 by the mapping $(D_1, D_2) : \mathbb{F}_p \times \mathbb{F}_p \to \mathbb{F}_p \times \mathbb{F}_p$.

In this section, we investigated the value set of the generalized Dickson polynomial, however, it still remains open to apply the generalized Dickson polynomial to solve the DLPwAI.

Remark 3.5.1. We can also analogously generalize this method to the n-variable Dickson polynomial. It results a map from $(\mathbb{F}_p)^n$ to itself, and the map would be d-to-1 on the $\mathrm{Irr}(a)$ (which will be defined similarly) of approximate size p^n/n, where $d|\Phi_{n+1}(p)$.

Chapter 4

Generalized DLP with Auxiliary Inputs

In this chapter, we define a new problem called the generalized DLPwAI (GDLPwAI). It is a problem to solve α for given $g^{\alpha^{e_1}}, \cdots, g^{\alpha^{e_d}}$, where $e_1, \cdots, e_d$ are arbitrary integers. The DLPwAI can be considered as the special case of the GDLPwAI with $e_i = i$. In this chapter, we propose an algorithm to solve the GDLPwAI when e_i's form a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$.

This chapter includes a joint work with Jung Hee Cheon and Yong Soo Song [14].

4.1 Multiplicative Subgroups of $\mathbb{Z}_n^{\times}$

Before the state of our main theorem, we introduce a new representation for multiplicative subgroup K of $\mathbb{Z}_n^{\times}$. From our observation, elements of a multiplicative subgroup $K \leq \mathbb{Z}_n^{\times}$ seem to form an arithmetic sequence in many cases.

4.1.1 Representation of a Multiplicative Subgroup of $\mathbb{Z}_n^\times$

Definition 4.1.1. For any positive integer n, let S be a subset of $\mathbb{Z}_n$. We define $\gcd(S; \mathbb{Z}_n)$ or $\gcd(S)$ unless confused, to be the greatest common divisor of all integers x such that $x \mod n$ belongs to S. Given a divisor λ of n, we define a subset K_λ of $\mathbb{Z}_n^\times$ by $K_\lambda := (1 + \lambda\mathbb{Z}_n) \cap \mathbb{Z}_n^\times$, where $1 + \lambda\mathbb{Z}_n := \{1 + \lambda m : m \in \mathbb{Z}_n\}$.

We can see that K_λ is a multiplicative subgroup of $\mathbb{Z}_n^\times$ because it is closed under the multiplication and inverse. If K is a multiplicative subgroup of $\mathbb{Z}_n^\times$, then K is a subgroup of K_λ for $\lambda = \gcd(K - 1)$ where $K - 1 = \{k - 1 : k \in K\} \subseteq \mathbb{Z}_n$.

Remark 4.1.1. For an even integer n and any multiplicative subgroup $K \leq \mathbb{Z}_n^\times$, every element of K is an odd integer so that $\gcd(K-1)$ is even. It shows that

$$K_\lambda = (1 + \lambda\mathbb{Z}_n) \cap \mathbb{Z}_n^\times = (1 + 2\lambda\mathbb{Z}_n) \cap \mathbb{Z}_n^\times = K_{2\lambda}$$

for odd λ. For this reason, we only treat the case that λ is even.

From now on, we restrict the case to $n = p-1$ for odd prime p. The next proposition determines the size of K_λ in $\mathbb{Z}_{p-1}^\times$ for given divisor λ of $p - 1$.

Proposition 4.1.1. *Let λ be a divisor of $p-1$. Then $|K_\lambda| = \frac{p-1}{\lambda} \cdot \prod_{q \in Q} \left(1 - \frac{1}{q}\right)$, where Q is the set of prime divisors of $p-1$ which do not divide λ. In particular, if $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$, then $|K_\lambda| = \phi(\frac{p-1}{\lambda})$, where ϕ denotes the Euler-totient function.*

Proof. Note that $1 + \lambda m \in K_\lambda$ if and only if $\gcd(1 + \lambda m, p - 1) = 1$, which is equivalent to $\gcd(1 + \lambda m, q) = 1$ for all $q \in Q$. Consider a surjective homomorphism

$$\begin{array}{rccc}\pi: & \mathbb{Z}_{p-1} & \longrightarrow & \mathbb{Z}_{\lambda}\times\mathbb{Z}_{q_1}\times\cdots\times\mathbb{Z}_{q_\ell} \\ & x & \longmapsto & (x \mod \lambda, x \mod q_1, \cdots, x \mod q_\ell)\,, \end{array}$$

where $Q=\{q_1,\cdots,q_\ell\}$. Then each element λm is in the set $K_\lambda - 1 \subseteq \mathbb{Z}_{p-1}$ if and only if $\pi(\lambda m)$ is contained in $\{0\}\times T$, where $T = (\mathbb{Z}_{q_1}\backslash\{-1\}) \times (\mathbb{Z}_{q_2}\backslash\{-1\})\times\cdots\times(\mathbb{Z}_{q_\ell}\backslash\{-1\})$. Hence

$$\begin{aligned} |K_\lambda| = |K_\lambda - 1| &= |\pi^{-1}(\{0\}\times T)| \\ &= |T|\cdot|\ker(\pi)| \\ &= \prod_{i=1}^{\ell}(q_i-1)\cdot\left(\frac{p-1}{\lambda\cdot\prod_{i=1}^{\ell}q_i}\right) \\ &= \frac{p-1}{\lambda}\cdot\prod_{i=1}^{\ell}\left(1-\frac{1}{q_i}\right) \end{aligned}$$

Moreover, if $\gcd\left(\lambda,\frac{p-1}{\lambda}\right)=1$, then Q is the set of all prime divisors of $\frac{p-1}{\lambda}$. Thus, we have $|K_\lambda| = \phi\left(\frac{p-1}{\lambda}\right)$. □

Proposition 4.1.2. *If λ is an even divisor of $p-1$, then* $\gcd(K_\lambda - 1;\mathbb{Z}_{p-1}) = \lambda$.

Proof. Let us use the same notations in the proof of Proposition 4.1.1. First, we note that an integer x such that $x \pmod{p-1} \in K_\lambda - 1 = \pi^{-1}(\{0\}\times T)$ is a multiple of λ, and $\gcd(K_\lambda-1;\mathbb{Z}_{p-1})$ is a multiple of λ by definition.

Let $P=\{p_j : 1\le j\le k\}$ be the set of common prime divisors of λ and $\frac{p-1}{\lambda}$. Then $P \,\dot{\cup}\, Q$ is the set of prime divisors of $\frac{p-1}{\lambda}$. Every element q of Q is greater than 2, and there exist integers m_i for $1\le i\le\ell$ satisfying $\lambda m_i \pmod{q_i}$ is not equal to 0 or -1. Using the Chinese Remainder Theorem, we can find an integer m such that $m\equiv m_i \pmod{q_i}$ for all $1\le i\le \ell$ and $m\equiv 1 \pmod{p_j}$ for all $1\le k\le j$.

We can check that $1+\lambda m$ is not divisible by $q\in Q$ and $1+\lambda m \pmod{p-1}$ is contained in K_λ. In addition, $\gcd(\lambda m;\mathbb{Z}_{p-1}) = \lambda\gcd(m;\mathbb{Z}_{\frac{p-1}{\lambda}}) = \lambda$ since

m is not divisible by every prime divisor of $\frac{p-1}{\lambda}$. Hence, $\gcd(K_\lambda - 1; \mathbb{Z}_{p-1})$ is equal to λ. $\square$

Example 4.1.1. *Consider a prime $p = 29$ and $\lambda = 4$ be an even divisor of $p-1$. Then, we have*

$$K_\lambda = K_4 = \{1, 5, 9, 13, 17, 21, 25\} \cap \mathbb{Z}_{28}^\times,$$

and 21 is the only element which is not in $\mathbb{Z}_{28}^\times$. Since $\frac{p-1}{\lambda} = 7$, we can see that the cardinality of K_4 is $\phi(7) = 6$ as shown in Proposition 4.1.1. Also we can check that $\gcd(K_4 - 1) = 4$.

4.2 A Group Action on $\mathbb{Z}_p^\times$

In this section, we consider a K-group action on $\mathbb{Z}_p^\times$ and partition $\mathbb{Z}_p^\times$ into disjoint orbits generated by group action. A group action on a set clearly induces a partition of the set with orbits. However, what we are dealing here is to partition $\mathbb{Z}_p^\times$ with only a few information. Namely, for a certain case, we can represent almost all elements of $\mathbb{Z}_p^\times$ with only two elements, one fixed point (*i.e.* an orbit with just one element) and the other point not a fixed point. We begin with defining the group action on $\mathbb{Z}_p^\times$.

Definition 4.2.1. Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^\times$. Define a $\underline{K\text{-action}}$ on $\mathbb{Z}_p^\times$ by $(k, x) \mapsto x^k$ for $k \in K$ and $x \in \mathbb{Z}_p^\times$. The $\underline{K\text{-orbit of } x}$ is a set $x^K := \{x^k : k \in K\}$. The $\underline{\text{set of fixed point } (\mathbb{Z}_p^\times)_K}$ is a set $\{x \in \mathbb{Z}_p^\times : x^k = x \text{ for all } k \in K\}$.

We can easily check that Definition 4.2.1 satisfies the definition of group action. Note that we have $|x^K| = |K|/|K_x|$ where K_x is a stabilizer of x which is a set defined by $K_x := \{k \in K : x^k = x\}$, thus $|x^K| = |K|$ if and only if

$|K_x| = 1$. The next proposition states that if two multiplicative subgroups H and K of $\mathbb{Z}_{p-1}^{\times}$ satisfies $\gcd(H-1) = \gcd(K-1)$, then the two sets of fixed points by H-action and K-action respectively are the same. Furthermore, the set of fixed points forms a cyclic group of order $\lambda = \gcd(H-1) = \gcd(K-1)$.

Proposition 4.2.1. *Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$ and $\lambda = \gcd(K-1)$. Then, $(\mathbb{Z}_p^{\times})_K = (\mathbb{Z}_p^{\times})_{K_\lambda} = \{z \in \mathbb{Z}_p^{\times} : z^{\lambda} = 1\}$.*

Proof. The set of fixed point by K-action is denoted by $(\mathbb{Z}_p^{\times})_K = \{z \in \mathbb{Z}_p^{\times} : z^{k-1} = 1 \text{ for all } k \in K\}$. Now it is easy to see that $z^{k-1} = 1$ for all $k \in K$ if and only if $z^{\lambda} = 1$ where $\lambda = \gcd\{k-1 : k \in K\}$. Since $\lambda = \gcd(K-1) = \gcd(K_\lambda - 1)$, we have $(\mathbb{Z}_p^{\times})_K = (\mathbb{Z}_p^{\times})_{K_\lambda}$ by the same argument. □

Let ξ be a primitive element in $\mathbb{Z}_p$, then $\zeta = \xi^{\frac{p-1}{\lambda}}$ is a generator of a cyclic group of fixed points $(\mathbb{Z}_p^{\times})_K = \langle\zeta\rangle = \{z \in \mathbb{Z}_p^{\times} : z^{\lambda} = 1\}$. Note that the orbit generated by $\zeta^i x$ satisfies $(\zeta^i x)^K = \zeta^i x^K$ for all $1 \le i \le \lambda$, since $\zeta^k = \zeta$ for all $k \in K$. The following proposition considers two orbits generated by $\zeta^i x$ and $\zeta^j x$ are disjoint for $0 \le i, j < \lambda$ and $i \ne j$.

Proposition 4.2.2. *(Disjoint Orbit Condition) Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$, ζ a generator of a cyclic group of fixed points $\{z \in \mathbb{Z}_p^{\times} : z^{\lambda} = 1\}$ for $\lambda = \gcd(K-1)$. If $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$, then two orbits $\zeta^i x^K$ and $\zeta^j x^K$ are disjoint i.e. $(\zeta^i x^K) \cap (\zeta^j x^K) = \emptyset$ for $0 \le i, j < \lambda, i \ne j$, and $x \in \mathbb{Z}_p^{\times}$.*

Proof. Note that two orbits are identical or disjoint. Suppose that $(\zeta^i x^K) \cap (\zeta^j x^K) \ne \emptyset$ for some i, j. Then, $\zeta^i x^K = \zeta^j x^K$ and $y := \zeta^{i-j} = x^{k_1 - k_2}$ for some $k := k_1 - k_2 \in K$. Since $(\zeta^{i-j})^{\lambda} = 1$ and $\left(x^{k_1-k_2}\right)^{\frac{p-1}{\lambda}} = 1$ for a non-fixed point $x \in \mathbb{Z}_p^{\times}$, the order of y divides both λ and $\frac{p-1}{\lambda}$. In other words, it divides $\gcd(\lambda, \frac{p-1}{\lambda})$ which equals to 1, following that y must be equal to 1. □

Example 4.2.1. *Let $K := K_4 = \{1, 5, 9, 13, 17, 25\} \leq \mathbb{Z}_{28}^{\times}$ and consider the K-action on $\mathbb{Z}_{29}^{\times}$. Then we have 4 disjoint orbits of length 6,*

$$\begin{aligned} 2^K &= \{2, 2^5, 2^9, 2^{13}, 2^{17}, 2^{25}\} = \{2, 3, 19, 14, 21, 11\} \\ 4^K &= \{4, 9, 13, 22, 6, 5\} \\ 7^K &= \{7, 16, 20, 25, 24, 23\} \\ 8^K &= \{8, 27, 15, 18, 10, 26\}, \end{aligned}$$

and 4 fixed points $\{1, 12, 17, 28\}$. Note that $1^4 \equiv 12^4 \equiv 17^4 \equiv 28^4 \equiv 1$ mod 29.

Since there is an one-to-one correspondence between $\zeta^i x^K$ and $\zeta^j x^K$ for all i, j, they have the same number of elements. If we define

$$\mathcal{O}_{x,K} := x^K \,\dot{\cup}\, \zeta x^K \,\dot{\cup} \cdots \dot{\cup}\, \zeta^{\lambda-1} x^K,$$

where $\dot{\cup}$ denotes the disjoint union, we have $|\mathcal{O}_{x,K}| = |x^K|\lambda$ for $x \in \mathbb{Z}_p^{\times}$. Along with the set of fixed points, we have $|\mathcal{O}_{x,K} \cup \langle\zeta\rangle| = (|x^K|+1)\lambda$ number of elements in $\mathbb{Z}_p^{\times}$ for a non-fixed point $x \in \mathbb{Z}_p^{\times}$. From now on, $\mathrm{ord}_p(x)$ denotes the order of x modulo p.

Remark 4.2.1. The set $\mathcal{O}_{x,K}$ behaves just like an extended orbit, which means that for $x, y \in \mathbb{Z}_p^{\times}$, $\mathcal{O}_{x,K}$ and $\mathcal{O}_{y,K}$ are disjoint or identical. In other words, $\mathcal{O}_{x,K} \cap \mathcal{O}_{y,K} \neq \emptyset$ implies $y = \zeta^i x^k$ and $\mathcal{O}_{x,K} = \mathcal{O}_{y,K}$. Therfore, $\mathbb{Z}_p^{\times}$ can be expressed by the disjoint union of distinct $\mathcal{O}_{x,K}$'s. Moreover, if $\mathcal{O}_{x,K} = \mathcal{O}_{y,K}$, then $y = \zeta^i x^k$ for some $0 \leq i < \lambda, k \in K$ and $y^\lambda = x^{\lambda k}$. It implies that $\mathrm{ord}_p(x^\lambda) = \mathrm{ord}_p(y^\lambda)$.

The next proposition gives a condition to satisfy $|x^K| = |K|$.

Proposition 4.2.3. *Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$, $\lambda = \gcd(K-1)$ and $x \in \mathbb{Z}_p$. If $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$, then $|x^K| = |K|$ for x satisfying*

$\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$. *In particular, if* $\frac{p-1}{\lambda}$ *is prime, then* $|x^K| = |K|$ *for* $x \notin (\mathbb{Z}_p^\times)_K$.

Proof. Note that $|x^K| = |K|$ if and only if $|K_x| = |\{k \in K : x^k = x\}| = 1$. Suppose that $x^k = x$ for some $k = 1 + \lambda n \in K$ and $0 \le n < \frac{p-1}{\lambda}$. It implies that $(x^\lambda)^n = 1$ for some $0 \le n < \frac{p-1}{\lambda}$. However, since $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$, n must be zero. It follows that K_x contains only one element, $k = 1$.

Since $(x^\lambda)^{\frac{p-1}{\lambda}} \equiv 1 \pmod p$ for all $x \in \mathbb{Z}_p$, we have $\mathrm{ord}_p(x^\lambda)$ divides $\frac{p-1}{\lambda}$. In addition, $\mathrm{ord}_p(x^\lambda) = 1$ if and only if $x \in (\mathbb{Z}_p^\times)_K$. Thus, if $\frac{p-1}{\lambda}$ is a prime, it follows that $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$ if and only if $x \notin (\mathbb{Z}_p^\times)_K$. □

Example 4.2.2. *Note that for* $p = 29$ *and* $\lambda = 4$, *we have* $|K| = |2^K| = |4^K| = |7^K| = |8^K| = 6$ *for* $K = K_4$, *and* $\langle 17 \rangle = \{17, 28, 12, 1\}$ *forms a cyclic group of fixed points. It is easily verified that* $17 \cdot 2^K = 4^K$, $28 \cdot 2^K = 8^K$ *and* $12 \cdot 2^K = 7^K$, *thus* $\mathcal{O}_{2,K} = 2^K \,\dot\cup\, 4^K \,\dot\cup\, 8^K \,\dot\cup\, 7^K = \mathbb{Z}_{29}^\times \backslash \langle 17 \rangle$.

The following proposition shows how many x's in $\mathbb{Z}_p^\times$ satisfy $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$.

Proposition 4.2.4. *Assume that* λ *is a divisor of* $p-1$. *Then there are exactly* $\lambda\phi(\frac{p-1}{\lambda})$ *elements* x *in* $\mathbb{Z}_p^\times$ *such that* $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$.

Proof. Let ξ be a primitive element of $\mathbb{Z}_p$. There exists a unique $0 \le j < p$ satisfying $x = \xi^j$ for any $x \in \mathbb{Z}_p^\times$. We will use the fact that $\mathrm{ord}_p(\xi^i) = \frac{p-1}{\gcd(i,p-1)}$ for all i.

From $\mathrm{ord}_p(x^\lambda) = \mathrm{ord}_p(\xi^{\lambda j}) = \frac{p-1}{\gcd(\lambda j, p-1)} = \frac{p-1}{\lambda}\frac{1}{\gcd(j,\frac{p-1}{\lambda})}$, we show that $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$ if and only if $\gcd(j, \frac{p-1}{\lambda}) = 1$. Therefore, there are exactly $\phi(\frac{p-1}{\lambda})$-number of j's modulo $\frac{p-1}{\lambda}$ satisfying $\gcd(j, \frac{p-1}{\lambda}) = 1$, thus $\lambda\phi(\frac{p-1}{\lambda})$-number of x's in $\mathbb{Z}_p^\times$ satisfying $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$. □

Note that $\lambda\phi(\frac{p-1}{\lambda}) = \lambda\frac{p-1}{\lambda}\prod_{q\in Q}(1-\frac{1}{q}) = (p-1)\prod_{q\in Q}(1-\frac{1}{q})$ where Q is the set of prime divisors of $\frac{p-1}{\lambda}$. Hence, if we randomly take x in $\mathbb{Z}_p^\times$, then the probability that $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$ is $\prod_{q\in Q}(1-\frac{1}{q})$. Moreover, if $\frac{p-1}{\lambda}$ has only large prime divisors, then the probability $\prod_{q\in Q}(1-\frac{1}{q})$ will be almost equal to 1.

Combining these results with Proposition 4.1.1, we surprisingly obtain an immediate partition of $\mathbb{Z}_p^\times$. Recall that for an even divisor λ of $p-1$, we defined a multiplicative subgroup $K_\lambda = \{1+\lambda n : n \in [0, \frac{p-1}{\lambda}) \cap \mathbb{Z}\} \cap \mathbb{Z}_{p-1}^\times$.

Theorem 4.2.1. *Let λ be an even divisor of $p-1$ satisfying $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$ and K_λ be a multiplicative subgroup of $\mathbb{Z}_{p-1}^\times$ defined as above. Consider the K_λ-action on $\mathbb{Z}_p^\times$. Let ζ be a generator of a cyclic group of fixed points by the K_λ-action, $\{z \in \mathbb{Z}_p^\times : z^\lambda = 1\}$. Then the followings hold:*

1. *If $\frac{p-1}{\lambda} = \mu$ is prime, then $\mathbb{Z}_p^\times = \mathcal{O}_{x,K_\lambda} \dot\cup (\mathbb{Z}_p^\times)_{K_\lambda}$ for $x \notin (\mathbb{Z}_p^\times)_{K_\lambda}$.*
2. *If $\frac{p-1}{\lambda} = \mu_1\cdots\mu_\ell$ is square-free for prime $\mu_1, \cdots, \mu_\ell$, then $\mathbb{Z}_p^\times = \dot\bigcup_{J\subseteq I} \mathcal{O}_{x^{\mu_J},K_\lambda}$ for $x \in \mathbb{Z}_p^\times$ such that $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$, where $I = \{1, 2, \cdots, \ell\}$ is an index set and $\mu_J = \prod_{j\in J}\mu_j$ for $J \subseteq I$ (For the convenience, define $\mu_\emptyset = 1$ for the empty subset $\emptyset \subseteq I$). In particular, $\mathcal{O}_{x^{\mu_I},K_\lambda} = (\mathbb{Z}_p^\times)_{K_\lambda}$.*

Proof. If $\frac{p-1}{\lambda} = \mu$ is prime, then $|K_\lambda| = \phi(\frac{p-1}{\lambda}) = \phi(\mu) = \mu - 1$ by Proposition 4.1.1. Note that $\mathcal{O}_{x,K_\lambda}$ and $(\mathbb{Z}_p^\times)_{K_\lambda}$ are disjoint subsets of $\mathbb{Z}_p^\times$ for $x \notin (\mathbb{Z}_p^\times)_{K_\lambda}$. Thus we have $|\mathcal{O}_{x,K_\lambda} \dot\cup (\mathbb{Z}_p^\times)_{K_\lambda}| = |\mathcal{O}_{x,K_\lambda}| + |(\mathbb{Z}_p^\times)_{K_\lambda}|$. By Proposition 4.2.3, we obtain $|\mathcal{O}_{x,K_\lambda}| = |x^{K_\lambda}|\lambda = |K_\lambda|\lambda = (\mu-1)\lambda$ and $|(\mathbb{Z}_p^\times)_{K_\lambda}| = \lambda$. Therefore, $|\mathcal{O}_{x,K_\lambda} \dot\cup (\mathbb{Z}_p^\times)_{K_\lambda}| = p-1$ deduces that $\mathcal{O}_{x,K_\lambda} \dot\cup (\mathbb{Z}_p^\times)_{K_\lambda} = \mathbb{Z}_p^\times$.

In the case that $\frac{p-1}{\lambda} = \mu_1\cdots\mu_\ell$ is square-free and $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$, we have $|x^{K_\lambda}| = |K_\lambda| = \phi(\frac{p-1}{\lambda}) = \phi(\mu_I) = \prod_{1\le j\le \ell}(\mu_j - 1)$ by Proposition 4.1.1. For a subset J of I and $y = x^{\mu_J}$, we first calculate $|y^{K_\lambda}|$ and $|\mathcal{O}_{y,K_\lambda}|$ by using the fact that $|y^{K_\lambda}| = |K_\lambda|/|(K_\lambda)_y|$, where $(K_\lambda)_y = \{k \in K_\lambda : y^k = y\}$. Since $k = 1+\lambda n \in (K_\lambda)_y$ if and only if $y^{k-1} = (x^{\mu_J})^{\lambda\cdot n} = 1$ if and only if $\mu_{I\setminus J} = \mu_I/\mu_J$ divides n, the size of $(K_\lambda)_y$ is equal to the number of n

satisfying that $1+\lambda n \in \mathbb{Z}_{p-1}^{\times}$, $0 \le n < \mu_I$ and $\mu_{I\setminus J}$ divides n. Therefore, by the similar argument in Proposition 4.1.1, we get

$$\begin{aligned} |(K_\lambda)_y| &= \left|\left\{n \in [0,\mu_I) \cap \mathbb{Z} : 1+\lambda n \in \mathbb{Z}_{p-1}^{\times} \text{ and } \mu_{I\setminus J} | (\lambda n)\right\}\right| \\ &= \left|\left\{n \in [0,\mu_I) \cap \mathbb{Z} : \mu_j \nmid (1+\lambda n) \text{ for each } j \text{ and } \mu_{I\setminus J} | n\right\}\right| \\ &= \frac{\mu_I}{\mu_{I\setminus J}} \cdot \prod_{j\in J}\left(1-\frac{1}{\mu_j}\right) \\ &= \mu_J \cdot \prod_{j\in J}\left(1-\frac{1}{\mu_j}\right) = \phi(\mu_J), \end{aligned}$$

resulting $|y^{K_\lambda}| = \frac{|K_\lambda|}{|(K_\lambda)_y|} = \frac{\phi(\mu_I)}{\phi(\mu_J)} = \phi(\mu_{I\setminus J})$ and $|\mathcal{O}_{y,K_\lambda}| = \lambda |y^{K_\lambda}| = \lambda\phi(\mu_{I\setminus J})$.

Since $\mathcal{O}_{x^{\mu_J},K_\lambda}$'s are pairwise disjoint for all $J \subseteq I$, we have $| \dot{\cup}_{J\subseteq I} \mathcal{O}_{x^{\mu_J},K_\lambda}| = \sum_{J\subseteq I} |\mathcal{O}_{x^{\mu_J},K_\lambda}| = \lambda \sum_{J\subseteq I} \phi(\mu_{I\setminus J})$. Finally, using elementary number theory, we have $\sum_{J\subseteq I} \phi(\mu_{I\setminus J}) = \sum_{d|\mu_I} \phi(d) = \mu_I$ and $| \dot{\cup}_{J\subseteq I} \mathcal{O}_{x^{\mu_J},K_\lambda}| = \lambda \cdot \mu_I = p-1$ deducing that $\mathbb{Z}_p^{\times} = \dot{\cup}_{J\subseteq I} (\mathcal{O}_{x^{\mu_J},K_\lambda})$. □

Note that for any given $x \in \mathcal{O}_{y,K_\lambda}$, there exist $0 \le i < \lambda$ and $k \in K_\lambda$ satisfying $x = \zeta^i y^k$. By virtue of Theorem 4.2.1, all elements in $\mathbb{Z}_p^{\times}$ can be expressed with only a few information. For example, we can simply partition $\mathbb{Z}_p^{\times}$ with only two elements $x \in \mathbb{Z}_p^{\times} - (\mathbb{Z}_p^{\times})_{K_\lambda}$ and $\zeta \in (\mathbb{Z}_p^{\times})_{K_\lambda}$, when $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$ and $q = \frac{p-1}{\lambda}$ is prime, so that any of element in $\mathbb{Z}_p^{\times}$ is of form $\zeta^i x^k$ for $0 \le i < \lambda$ and $k \in K$. In our example, with only $x = 2$ and $\zeta = 17$, we can express all elements in $\mathbb{Z}_{29}^{\times}$.

In the case of $\frac{p-1}{\lambda} = \mu_1 \cdots \mu_\ell$ is square-free and $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$, Remark 4.2.1 says that $\mathrm{ord}_p(y^\lambda) = \mu_{I\setminus J}$ if $y \in \mathcal{O}_{x^{\mu_J},K_\lambda}$. The converse is also true because $\mathbb{Z}_p^{\times} = \dot{\cup}_{J\subseteq I} \mathcal{O}_{x^{\mu_J},K_\lambda}$ and y cannot be contained in $\mathcal{O}_{x^{\mu_{J'}},K_\lambda}$ for $J \ne J' \subseteq I$.

4.3 Polynomial Construction

In this section, we will define a polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree d having small value sets. Recently, the similar idea was developed by Kim and Cheon [37] to solve the DLPwAI. Their approach exploited the fast multipoint evaluation method, so the degree of their polynomial was restricted to at most $d \approx p^{1/3}$ due to the efficiency issue.

The polynomial we will use in this paper is of very large degree which might be greater than $p^{1/3}$ but is sparse (all but d coefficients are zero) and have small value sets. Thus the fast multipoint evaluation method as in [37] seems hardly to be applied in our case. Instead, we take somewhat different approach with the idea developed in Section 4.2. We will define a polynomial so that it takes the same value for all elements in an orbit. In the proof of our main theorem, we will make some lists of $f(\alpha_1), \cdots, f(\alpha_\ell)$ from $f(\alpha)$ where α_i's are the representatives of distinct orbits and α is a discrete log to find. Then we find an index j such that $f(\alpha_j) = f(\beta)$ for randomly chosen $\beta \in \mathbb{Z}_p^\times$ *i.e.* we find an orbit in which β is contained. For this process, $f(\alpha)$ should be nonzero.

Definition 4.3.1. Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^\times$. Define a polynomial $f_K(x)$ over $\mathbb{Z}_p$ by $f_K(x) := \sum_{k\in K} x^k$. We will simply write $f_K = f$ if there is no ambiguity in the meaning.

By the definition, it is clear that f_K takes the same value for the elements in the same orbit defined by K-action.

Proposition 4.3.1. *For any $k \in K$ and $x \in \mathbb{Z}_p^\times$, we have $f(x^k) = f(x)$. If $\zeta^i \in (\mathbb{Z}_p^\times)_K$ is a fixed point, then $f(\zeta^i x) = \zeta^i f(x)$.*

Since the degree of $f = f_K$ might be large (approximately p), it looks hard to evaluate $f(\alpha_1), \cdots, f(\alpha_\ell)$ in $O(\ell)$ time complexity for random α_i's with fast multipoint evaluation method. However, for a non-fixed point $\alpha \in \mathbb{Z}_p^\times$ and a fixed point (not necessarily generator) $\zeta \in (\mathbb{Z}_p)_K$, we can compute $f(\alpha), f(\zeta\alpha) = \zeta f(\alpha), \cdots, f(\zeta^\ell \alpha) = \zeta^\ell f(\alpha)$ in ℓ multiplications by ζ with $O(|K|)$ exponentiations for computing $f(\alpha)$. Furthermore, if $f(\alpha)$ is nonzero, then we can deduce that all $\alpha, \zeta\alpha, \cdots, \zeta^\ell\alpha$ are the different representatives for distinct orbits. The following proposition calculates $f(x)$ explicitly in special cases.

Proposition 4.3.2. *Assume that λ is an even divisor of $p-1$ satisfying* $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$. *Let $K = K_\lambda$ and $f = f_K$ be defined as aforementioned. Then the followings hold:*

1. *If $\frac{p-1}{\lambda} = \mu$ is prime, then $f(x) \neq 0$ for $x \in \mathbb{Z}_p^\times$.*

2. *If $\frac{p-1}{\lambda} = \mu_1 \cdots \mu_\ell$ is square-free for prime $\mu_1, \ldots, \mu_\ell$, then $f(x) \neq 0$ for $x \in \mathbb{Z}_p^\times$.*

Proof. If $\frac{p-1}{\lambda} = \mu$ is prime, then $|K| = \mu - 1$ by Proposition 4.1.1. Consider a map from $\mathbb{Z}_\mu$ to itself defined by $n \mapsto (1 + \lambda n)$. Since λ and μ are relatively prime, this map is bijective. In other words, $1 + \lambda n$ for $0 \leq n < \mu$ induces complete residue modulo μ. Thus, there exists a unique $0 \leq n_0 < \mu$ such that $1 + \lambda n_0$ is divisible by μ. Therefore,

$$f(x) = \sum_{k \in K} x^k = \sum_{0 \leq n < \mu} x^{1+\lambda n} - x^{1+\lambda n_0} = x \cdot \frac{x^{p-1} - 1}{x^\lambda - 1} - x^{1+\lambda n_0} = -x^{1+\lambda n_0}$$

for $x \notin (\mathbb{Z}_p^\times)_K$. Otherwise, if $x^\lambda = 1$ then $x^k = x$ for all $k \in K$ and $f(x) = (\mu - 1)x \neq 0$.

In the case of $\frac{p-1}{\lambda} = \mu_1 \cdots \mu_\ell$ is square-free, $|K| = \phi(\mu_1 \cdots \mu_\ell)$ by Proposition 4.1.1. By similar argument as above, for a subset J of an index set $I = \{1, 2, \cdots \ell\}$, let $\mu_J = \prod_{j \in J} \mu_j$, and define a map from $\mathbb{Z}_{\mu_J}$ to itself by

$n \mapsto (1+\lambda n)$. Since λ and μ_J are relatively prime, it also induces the complete residue modulo μ_J. Thus, there exists a unique $0 \le n_J < \mu_J$ such that $1 + \lambda n_J$ is divisible by μ_J (For convenience, define $\mu_J = 1$ and $n_J = 0$ for empty set $J = \emptyset$). We easily check that $n_J \equiv n_I \pmod{\mu_J}$ for all $J \subseteq I$. Now, $\mathrm{ord}_p(x^\lambda) = \mu_{I_0}$ for some $I_0 \subseteq I$ since $\mathrm{ord}_p(x^\lambda)$ is a divisor of $\frac{p-1}{\lambda} = \mu_I$. For $J \subseteq I$, $x^{\lambda\mu_J} = 1$ if and only if $I_0 \subseteq J$.

Using the inclusion–exclusion principle, we have

$$f(x) = \sum_{k\in K} x^k = \sum_{J\subseteq I} (-1)^{|J|} \sum_n x^{1+\lambda n},$$

where n in summation runs through $0 \le n < \mu_I$ satisfying $n \equiv n_J \pmod{\mu_J}$.

If $I_0 \nsubseteq J \subseteq I$, then $x^{\lambda\mu_J} \ne 1$ and $\sum_n x^{1+\lambda n} = x^{1+\lambda n_J}\frac{x^{p-1}-1}{x^{\lambda\mu_J}-1} = 0$. Otherwise $I_0 \subseteq J \subseteq I$, then $x^{\lambda\mu_J} = 1$ and $\sum_n x^{1+\lambda n} = \sum_n x^{1+\lambda n_J} = \frac{\mu_I}{\mu_J} x^{1+\lambda n_J} = \mu_{I\setminus J} x^{1+\lambda n_I}$ since n in summation is equivalent to n_J modulo μ_J, and $n_J \equiv n_I \pmod{\mu_J}$.

Finally, we have

$$\begin{aligned}
f(x) &= \sum_{J\subseteq I} (-1)^{|J|} \sum_n x^{1+\lambda n} = \sum_{I_0\subseteq J\subseteq I} (-1)^{|J|} \sum_n x^{1+\lambda n} \\
&= x^{1+\lambda n_I} \sum_{I_0\subseteq J\subseteq I} (-1)^{|J|} \mu_{I\setminus J} = x^{1+\lambda n_I} \sum_{J\subseteq I\setminus I_0} (-1)^{|I\setminus J|} \mu_J \\
&= x^{1+\lambda n_I} (-1)^\ell \prod_{j\in I\setminus I_0} (1-\mu_j) \ne 0.
\end{aligned}$$

In particular, if $\mathrm{ord}_p(x^\lambda) = \mu_I$, then $f(x) = (-1)^\ell x^{1+\lambda n_I}$. □

The above proposition says that $f_K(x)$ is not identically zero for $K_\lambda = K$ for even divisor λ of $p-1$. Actually, it appears to be of form $f_K(x) = -x^d$ where $\gcd(d, p-1)$ is large, however in our application, it is desirable that $f_K(x) \ne 0$ but is not of simple form such as x^d, where d has large common divisor with $p-1$, since this simple form leads us to the already known

Cheon's $p-1$ algorithm. In many cases, for a non proper subgroup K of K_λ, $f_K(x)$ also tends to not to be identically zero, although it seems hard to show it.

Example 4.3.1. *For $K = K_4 = \{1, 5, 9, 13, 17, 25\} \leq \mathbb{Z}_{28}^\times$, define $f_K(x) = x+x^5+x^9+x^{13}+x^{17}+x^{25} = -x^{21} \in \mathbb{Z}_{29}[x]$, where 21 and 28 have common divisor 7. For a subgroup $K' = \langle 9 \rangle = \{9, 25, 1\}$ of K, we have $K/\langle 9 \rangle = \{1, 5\}$. Now consider $f_{K'}(x) = x + x^9 + x^{25}$. Then $f_{K'}(x)$ takes same value for x in the same orbit. We have 8 disjoint orbits of length 3 and 4 fixed points. Note that the fixed points for K and K' are same as shown in Proposition 4.2.1.*

$$\begin{aligned} 2^{K'} &= \{2, 19, 11\}, & 2^{5K'} &= 3^{K'} = \{3, 14, 21\} \\ 4^{K'} &= \{4, 13, 5\}, & 4^{5K'} &= 9^{K'} = \{9, 22, 6\} \\ 7^{K'} &= \{7, 20, 23\}, & 7^{5K'} &= 16^{K'} = \{16, 25, 24\} \\ 8^{K'} &= \{8, 15, 26\}, & 8^{5K'} &= 27^{K'} = \{27, 18, 10\}. \end{aligned}$$

The polynomial $f_{K'}(x)$ takes nonzero value $2 + 19 + 11 \equiv 3 \mod 29$ for all $x \in 2^{K'}$, and we can check that $f_{K'}(x)$ take distinct values for disjoint orbits.

Proposition 4.3.3. *Assume that λ is an even divisor of $p-1$ satisfying $\gcd(\lambda, \frac{p-1}{\lambda}) = 1$. Let $K = K_\lambda$ and $f = f_K$. If $\frac{p-1}{\lambda} = q^e$ for some prime q and $e \geq 2$, then $f(x) = 0$ unless $x^{\lambda q} = 1$ in $\mathbb{Z}_p^\times$.*

Proof. Since $\frac{p-1}{\lambda}$ has only one prime divisor q, we can efficiently express elements of K and compute $f(x)$. For $n \in \mathbb{Z}_\mu$, $1 + \lambda n$ is contained in K if and only if $\gcd(1 + \lambda n, q) = 1$. Since $1 + \lambda n \equiv 0 \pmod q$ has exactly one solution $n_0 \equiv -\lambda^{-1}$ in modulo q, there exist q^{e-1}-number of solutions $\{n_0 + qm : 0 \leq m < q^{e-1}\}$ in $\mathbb{Z}_\mu$. Therefore, $f(x)$ is computed by

$$\begin{aligned} f(x) &= \sum_{n\in[0,\frac{p-1}{\lambda})\cap\mathbb{Z},1+\lambda n\in K} x^{1+\lambda n} = \sum_{0\le n<q^e} x^{1+\lambda n} - \sum_{0\le m<q^{e-1}} x^{1+\lambda(n_0+qm)} \\ &= x\left(\sum_{0\le n<q^e} x^{\lambda n}\right) - x^{1+\lambda n_0}\left(\sum_{0\le m<q^{e-1}} x^{\lambda qm}\right), \end{aligned}$$

and it is equal to zero unless $x^{\lambda q} = 1$. However, there are only $\lambda q = \frac{p-1}{q^{e-1}}$-number of such elements x in $\mathbb{Z}_{p-1}^\times$. □

In general, if $\frac{p-1}{\lambda}$ is not square-free, then $f_{K_\lambda}(x) = 0$ for most of the elements in $\mathbb{Z}_{p-1}^\times$. Modifying the proofs of Proposition 4.3.2 and Proposition 4.3.3 easily show it. We will omit details here.

4.4 Main Theorem

By using a group action on $\mathbb{Z}_p^\times$, we can efficiently partition $\mathbb{Z}_p^\times$ with only a few elements. This leads us to a new algorithm that solves the GDLPwAI efficiently. Now we can state our main theorem as follows.

Theorem 4.4.1. *Let K be a multiplicative subgroup of $\mathbb{Z}_{p-1}^\times$ with $\lambda = \gcd(K-1)$. Assume that we are given $\left\{\left(k, g^{\alpha^k}\right) : k \in K\right\}$ and $|\alpha^K| = |K|$. Then, we can solve $\alpha \in \mathbb{Z}_p$ in $O\left(\frac{p}{\lambda}\right)$ exponentiations in $\mathbb{Z}_p$ and $O\left(\frac{p}{|K|\sqrt{\lambda}} + |K|\right)$ exponentiations in G unless $\sum_{k\in K} \alpha^k = 0$.*

Proof. We give a sketch of the proof following the next steps.

1. For given g^{α^k} for all $k \in K$, one computes $g^{f(\alpha)} = \prod_{k\in K} g^{\alpha^k} \in G$ in $|K|$ multiplications in G. Note that $g^{f(\alpha)} \neq 1$, since $f(\alpha) \neq 0$.

2. Take a random element β from $\mathbb{Z}_p^\times$ and compute $f(\beta) = \sum_{k\in K} \beta^k \in \mathbb{Z}_p$ in $|K|$ exponentiations in $\mathbb{Z}_p$. If $\beta \in \mathcal{O}_{\alpha,K}$, then there exists a unique $0 \le t < \lambda$ satisfying $\alpha^K = \zeta^t \beta^K$ and $f(\alpha) = \zeta^t f(\beta)$.

3. To find such t, we use Baby-Step Giant-Step method. Let $L := \lceil\sqrt{\lambda}\rceil$. Make two lists $\{g^{f(\zeta^{L\cdot i}\beta)} = (g^{f(\beta)})^{\zeta^{L\cdot i}} \in G : 0 \le i < L\}$ and $\{g^{f(\zeta^{-j}\alpha)} = (g^{f(\alpha)})^{\zeta^{-j}} \in G : 0 \le j < L\}$ in $2\sqrt{\lambda}$ exponentiations in G. If $\beta \in \mathcal{O}_{\alpha,K}$, these two lists must have a collision since there exist $0 \le i, j < L$ satisfying $t = Li + j$.

4. Repeat the steps 2 and 3 until finding a collision. The expected number of repetitions is $\frac{p}{|K|\lambda}$, since the probability that $\beta \in \mathcal{O}_{\alpha,K}$ is $\frac{|\mathcal{O}_{\alpha,K}|}{p} = \frac{|\alpha^K|\lambda}{p} = \frac{|K|\lambda}{p}$.

5. Locate $g^{\zeta^t\beta}$ from the set $\left\{g^{\alpha^k} : k \in K\right\}$ to find $k_0 \in K$ such that $g^{\alpha^{k_0}} = g^{\zeta^t\beta}$. This gives $\alpha = (\zeta^t\beta)^{k_0^{-1}}$ in $|K|$ comparisons in G.

We carry out the above process in $|K|$ multiplications in G in Step 1, $O\left(\frac{p}{|K|\lambda} \cdot |K|\right) = O\left(\frac{p}{\lambda}\right)$ exponentiations in $\mathbb{Z}_p$ in Step 2 and $O\left(\frac{p}{|K|\sqrt{\lambda}}\right)$ exponentiations in G in Step 3 and 4, and $|K|$ comparisons in G in Step 5. The overall complexity is as in the theorem. □

Remark 4.4.1. In the proof of Theorem 4.4.1, we may find a fake collision. That is, some element $\beta \in \mathbb{Z}_p$ could satisfy $f(\alpha) = \zeta^t f(\beta)$ but $\zeta^t\beta \notin \alpha^K$. If a fake collision occurs in Step 3 and 4, there would be no element $k_0 \in K$ such that $\alpha^{k_0} = \zeta^t\beta$ and we can check it in Step 5. They do not affect the total complexity.

For any multiplicative subgroup K of $\mathbb{Z}_{p-1}^\times$, K is a multiplicative subgroup of K_λ where $\lambda = \gcd(K-1)$. Hence we can define $\kappa = [K_\lambda : K]$.

Corollary 4.4.1. *For a multiplicative subgroup K of $\mathbb{Z}_p^\times$, set $\lambda = \gcd(K-1)$ and define $\kappa = [K : K_\lambda]$. Assume that the computational cost for the multiplications in G is a constant times of the cost for the multiplications in $\mathbb{Z}_p$. Then we can solve the GDLPwAI in $O\left(\left(\kappa\sqrt{\lambda} + \frac{p}{\lambda}\right)\log p\right)$ multiplications in $\mathbb{Z}_p$.*

Proof. In Proposition 4.1.1, we showed that $|K_\lambda| = \frac{p-1}{\lambda}\prod_{q\in Q}(1-\frac{1}{q})$ where Q is the set of prime divisors of $p-1$ not dividing λ. We may assume that $\prod_{q\in Q}(1-\frac{1}{q})$ is a constant greater than zero since $\prod_{q\in Q}(1-\frac{1}{q}) \geq \frac{\phi(\frac{p-1}{\lambda})}{\frac{p-1}{\lambda}} \geq \frac{1}{6\log\log\frac{p-1}{\lambda}}$ and $\log\log\frac{p-1}{\lambda}$ is not so large for usual size of p. In fact, $\prod_{q\in Q}(1-\frac{1}{q})$ is much greater than this lower bound in almost cases. Then we have $|K| = \frac{|K_\lambda|}{\kappa} = O\left(\frac{p}{\lambda\kappa}\right)$ and $\frac{p}{|K|\sqrt{\lambda}} = O\left(\kappa\sqrt{\lambda}\right)$.

By Theorem 4.4.1, the overall complexity is $O(|K|\log p) = O\left(\frac{p}{\lambda}\log p\right)$ multiplications in $\mathbb{Z}_p$ and $O\left(\left(|K| + \frac{p}{|K|\sqrt{\lambda}}\right)\log p\right) = O\left(\left(\kappa\sqrt{\lambda} + \frac{p}{\lambda}\right)\log p\right)$ multiplications in G. By the assumption, we can put them together in one notation. $\square$

Example 4.4.1. *Consider a multiplicative group $\mathbb{Z}_q^\times$ for prime $q = 1984044749$. The element $g = 268435456 \in \mathbb{Z}_q^\times$ generates the multiplicative subgroup $G = \langle g\rangle$ of 20-bit prime order $p = 70858741$. Suppose that we are given $\left\{\left(k, g^{\alpha^k}\right) : k \in K\right\} = \{(1, 368141755), (9447833, 908277040), (14171749, 1018628336), (51963077, 651549246)\}$ for the multiplicative subgroup K of $\mathbb{Z}_{p-1}^\times$ with $\lambda = \gcd(K;\mathbb{Z}_{p-1}) = 4723916$. Following Theorem 4.4.1, we have $g^{f(\alpha)} = 104646375$ and $f(\beta) = 29994755$ for randomly chosen $\beta = 27015355$ in G. Using the BSGS technique, we find $t = 993142$ satisfying $g^{f(\alpha)} = g^{\zeta^t f(\beta)}$ for a primitive element ξ and a fixed point $\zeta = \xi^{\frac{p-1}{\lambda}}$. Then we find out that $\alpha^{k_0} = \zeta^t\beta$ for $k_0 = 51963077$ by comparing $g^{\zeta^t\beta}$ with $\{g^{\alpha^k} : k \in K\}$. Finally, we have $\alpha = (\zeta^t\beta)^{k_0^{-1}} = 37217684$.*

Example 4.4.2. *We use the same notations with Example 4.4.1. Set $q = 8307519720650407$, $g = 3814697265625 \in \mathbb{Z}_q^\times$. The element g has the order $p = 461528873369467$ of 50-bit prime. We are given our instance for a multiplicative subgroup K of K_λ such that $\lambda = 4742043558, |K_\lambda| = 97326, |K| = 16221$. Our algorithm finds that*

$$\alpha = \zeta^t\beta = 55526261320836$$

for $\zeta = 265871590696697$, $\beta = 257387303120427$ *and* $t = 275438533$.

In summary, if we are given g^{α^k} for all $k \in K_\lambda$, then $\kappa = 1$ and we can solve the GSDL problem in $O\left(\left(\sqrt{\lambda} + \frac{p}{\lambda}\right)\log p\right)$. However, in this case, $g^{f_{K_\lambda}(\alpha)} = g^{-d}$ with nontrivial $\gcd(d, p-1)$, which falls into the Cheon's $p-1$ algorithm. When we are working with $|K| < |K_\lambda|$, then we need to carry out $O\left(\left(\kappa\sqrt{\lambda} + \frac{p}{\lambda}\right)\log p\right)$ multiplications, so we want $\kappa > 1$ to be sufficiently small. The computation amount can be reduced to $O\left(p^{1/3}\log p\right)$, when κ is small enough and $\lambda \approx p^{2/3}$.

Remark 4.4.2. If we assume that α is chosen randomly in $\mathbb{Z}_p^\times$, the condition $|\alpha^K| = |K|$ is satisfied with high probability. As we mentioned in Proposition 4.2.3 and Proposition 4.2.4, there are $\lambda\phi(\frac{p-1}{\lambda})$-number of x's in $\mathbb{Z}_p^\times$ such that $\mathrm{ord}_p(x^\lambda) = \frac{p-1}{\lambda}$, and they satisfy $|x^K| = |K|$. Therefore, the probability is greater than $\frac{1}{6\log\log(p-1)}$, since $\frac{\lambda\phi(\frac{p-1}{\lambda})}{p-1} \geq \frac{\phi(p-1)}{p-1}$ and $\frac{\phi(n)}{n} \geq \frac{1}{6\log\log n}$ for all $n \geq 5$.

Remark 4.4.3. It is hard to compute the probability of $\sum_{k\in K}\alpha^k = 0$ in general, but we can predict that $f_K(x) = 0$ has not so many roots in $\mathbb{Z}_p$ if $\frac{p-1}{\lambda}$ is a square-free which is relatively prime to λ. Let $\kappa = [K_\lambda : K]$ and $\{k_1, \cdots k_\kappa\}$ be elements of distinct left cosets of K in K_λ. Then we have $f_{K_\lambda}(x) = \sum_{i=1}^{\kappa} f_K(x^{k_i})$. We saw in Proposition 4.3.2 that if $\frac{p-1}{\lambda}$ is a square-free which is relatively prime to λ, then f_{K_λ} is a monomial and hence it is never zero on $\mathbb{Z}_p$. Therefore, we can say that the condition $f_K(\alpha) \neq 0$ in Theorem 4.4.1 is not so unnatural in this case. In the contrary, it may be harder to satisfy the condition $f_K(\alpha) \neq 0$ if $\frac{p-1}{\lambda}$ has prime powers. The case of Proposition 4.3.3 is a typical example.

We have another strategy to avoid 'bad cases' aforementioned by randomizing α. In the case of $|\alpha^K| \neq |K|$, take a random element γ in $\mathbb{Z}_p^\times$

and compute new parameters $\{(g^{\alpha^k})^{\gamma^k} : k \in K\}$, which can be done in $|K|$ exponentiations in $\mathbb{Z}_p$ and G. We repeat this process until finding γ which satisfies $|(\alpha\gamma)^K| = |K|$, and the expected number of repetition is less than $6 \log\log(p-1)$. Finally, we can compute $\alpha\gamma$ in $O\left(\frac{p}{\lambda|K|}(\sqrt{\lambda} + |K|)\right)$ exponentiations by Theorem 4.4.1, and get $\alpha = (\alpha\gamma) \cdot \gamma^{-1}$. The total number of computations is $O\left(|K| \log\log p + \frac{p}{\lambda|K|}(\sqrt{\lambda} + |K|)\right)$, which does not have significant difference with $O\left(\frac{p}{\lambda|K|}(\sqrt{\lambda} + |K|)\right)$.

This strategy can be also used in the case of $f_K(\alpha) = 0$. We can compute new parameters $\{(g^{\alpha^k})^{\gamma^k} : k \in K\}$ in $|K|$ exponentiations in $\mathbb{Z}_p$, and check whether $f_K(\alpha\gamma)$ is equal to zero or not in $|K|$ multiplications in G. The expected number of repetition depends on the number of roots of $f_K(x) = 0$ in $\mathbb{Z}_{p-1}$. This algorithm must be more efficient than the above, but the exact complexity is not resolved yet.

Chapter 5

The Pairing Inversion Problem

5.1 Introduction

A pairing $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map from two additive groups G_1 and G_2 to a multiplicative group G_T. A bilinearity means $e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q)$ and $e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2)$, where P_1, P_2 and $P \in G_1$ and Q_1, Q_2 and $Q \in G_2$. A non-degenracy means that $e(P, Q) = 1$ implies $P = 0$ or $Q = 0$.

The pairing is staple in the public-key cryptography: It has been used to construct the cryptosystems with various functionalities, for example, the identity-based encryption schemes [6], the one-round three partite key exchange protocol [30] and the broadcast encryptions [7], etc.

The security of the pairing-based cryptography relies on the hardness of the pairing inversion problem which is required to solve Q (or, P) from the value of $e(P, Q)$ and P (or, Q). And if one can solve the pairing inversion problem in a polynomial time, then it is possible to solve the DLP since the algorithm solving the pairing inversion gives a solution for the computational Diffie-Hellman problem. From this point, we consider the pairing inversion problem as a kind of the auxiliary informations which can be used to solve the DLP. The pairing inversion problem is also considered by several researches [22, 46, 33, 36, 10].

In the cryptographic area, it is widely used the Weil pairing and Tate pairing, both of them are defined on the elliptic curve groups over the finite fields. For the efficiency issues, the Tate pairing is often desirable. Therefore, in this context we concentrate our concern to invert the Tate pairing.

Let $E(\mathbb{F}_{q^k})$ be an elliptic curve defined over $\mathbb{F}_{q^k}$ for prime power q. The value of the Tate pairing at $(P, Q) \in E(\mathbb{F}_{q^k}) \times E(\mathbb{F}_{q^k})$ is given by $e(P, Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}}$ where $f_{r,P} \in \mathbb{F}_{q^k}[x, y]$, which is called the Miller function, and k, which is the smallest positive integer satisfying $r|q^k - 1$ (we call such k by the embedding degree). The Tate pairing can be computed within the sequential two steps: first, one computes $f_{r,P}(Q)$ using the Miller's algorithm [41] and then we finalize the computation by powering of $\frac{q^k-1}{r}$. Each step is called the Miller step and the final exponentiation step, respectively.

The naive approach for the Tate pairing inversion is to invert the final exponentiation step (EI: exponent inversion) and then invert the Miller step (MI: Miller inversion). The recent works by Kanayama and Okamoto [33] and Chang et al. [10] showed that the pairing inversion problem on the ate pairings [27, 55, 57], variants of the Tate pairing, reduces to the exponent inversion problem. In [10], they gave the complexity of the Miller inversion for the optimal pairing [55]. In [54], Vercauteren showed that the complexity of the exponent inversion problem in the ate pairing is related to the sum of the absolute values of the coefficients of the exponent in the q-ary representation.

In this chapter, we aim our concern to reduce the complexity of the final exponentiation step encompassing the (non-)parameterized family of the pairing-friendly curves. This chapter includes a part of the joint work with Sungwook Kim and Jung Hee Cheon [38].

In the Tate pairing (or, its variants such as the ate pairing), the final exponentiation is to raise to the power by $(q^k - 1)/r$. For even k, we split the exponent into three parts

$$(q^k - 1)/r = [(q^{k/2} - 1)/r] \cdot [(q^{k/2} + 1)/\Phi_k(q)] \cdot [\Phi_k(q)/r],$$

where $\Phi_k(x)$ is the k-th cyclotomic polynomial. With the Frobenius map, the former two parts can be computed efficiently. Thus the powering by $\lambda := \Phi_k(q)/r$ is the hard part of the computation. Consider the well-known exponentiation method, the multi-exponentiation technique. When we write the hard part of the exponent as $\lambda = \lambda_0 + \lambda_1 q + \cdots + \lambda_{\varphi(k)-1} q^{\varphi(k)-1}$ in the q-ary representation, the multi-exponentiation with the width w computes the exponentiation by λ with $\log_2 q$ squarings and $(\log_2 q)/w + 2^{w\varphi(k)}$ multiplications and $O(2^{w\varphi(k)})$ storage. Throughout this chapter, we mainly focus on reducing the size of $\max_i |\lambda_i|$, since it is closely related to the number of squarings. We also define this value by $\kappa(\lambda)$.

With the assumption that λ behaves like the random integer for the random curves, the value of $\kappa(\lambda)$ is expected to have $\log_2 q$, however, interestingly, we note that for most existing parameterized families of pairing-friendly curves the value $\kappa(\lambda)$ is much less than $\log_2 q$. For example, it is about $(\log_2 q)/2$ for the supersingular curves with embedding degree $k = 6$ and $3(\log_2 q)/4$ for the BN curves [3] which has the embedding degree $k = 12$. One can observe that these values satisfy $\kappa = \left(1 - \frac{1}{\rho\varphi(k)}\right) \log_2 q$, surprisingly it is not a coincidence, we shall show that this value is the optimal for any pairing-friendly curves.

Summarizing our goal in the twofold, we first investigate *when* the parameterized families of pairing-friendly curves have small $\kappa(\lambda)$'s and *what* is the optimal value for this. For the second, we give an universal approach to attain the optimal value of the $\kappa(\lambda)$ for any pairing-friendly curves particularly encompassing non-parameterized pairing-friendly curves.

Our contributions

Consider pairing-friendly curves in which q and r are parameterized by polynomials $q(x)$ and $r(x)$ in $\mathbb{Q}[x]$, and write the final exponent $\lambda(x) := \Phi_k(q(x))/r(x)$ as $\lambda_0(x) + \lambda_1(x)q(x) + \cdots + \lambda_{\varphi(k)-1}q(x)^{\varphi(k)-1}$ with $\lambda_i(x) \in \mathbb{Q}[x]$ for all $i = 0, 1, \ldots, \varphi(k) - 1$. We show that all known construction methods of parameterized pairing-friendly elliptic curves satisfy $\kappa(\lambda) \geq \left(1 - \frac{1}{\rho\varphi(k)}\right) \log_2 q$. The equality holds when each leading coefficients of $q(x)$ and $\lambda_i(x)$ are small and $\max_i\{\deg(\lambda_i(x))\} = \deg(q(x)) - \deg(r(x))/\varphi(k)$.

Next, we propose a method to obtain a modified pairing with small κ for any pairing-friendly elliptic curves. More precisely, our method uses lattice reduction to find an integer m with $\gcd(m, r) = 1$ such that $\kappa(m\lambda) = \frac{1}{\varphi(k)} \log_2 (\Phi_k(q)/r)$, which is about $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log_2 q$. When using a modified Tate pairing $\bar{e}(P, Q) := e(P, Q)^m$, we can reduce the number of squarings in the final exponentiation by a factor of $\left(1 - \frac{1}{\rho\varphi(k)}\right)$ from the usual Tate pairing. We remark that similar idea to use this modified pairing has been also used in [19]. The work in [19] focuses on reducing the coefficients of $\lambda_i(x)$'s in Scott et al's technique [49] for parameterized family of curves. Our method works for arbitrary pairing-friendly curves even when Scott et al.'s method is not applicable. Furthermore, we find the optimality of complexity for final exponentiation step. We show that $\kappa(m\lambda)$ is lower-bounded by $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log_2 q - \log_2 \varphi(k)$ for any integer m with $\gcd(m, r) = 1$. It is interesting that this bound almost equals to the lower bound in the first part. We verify our argument by applying it to the DEM curves [17], Cocks-Pinch curves [16].

Outline of the paper

This paper is organized as follows. In Section 5.2.1, we briefly introduce some backgrounds of pairings, pairing-friendly curves, and exponentiation method we use to analyze the number of squarings in the final exponentiation step. In Section 5.3.1, we give the analysis on parameterized families of pairing-friendly curves in the sense of the final exponentiation-efficiency. In Section 5.3.2, we propose a general method to accelerate the final exponentiation and show the number of squarings in the final exponentiation is bounded below by $\left(1 - \frac{1}{\rho\varphi(k)}\right)\log_2 q$. We present examples in Section 5.3.3 and finally conclude in Section **??**.

5.2 Preliminaries

Throughout this paper, we denote $\log_2(\cdot)$ by $\log(\cdot)$.

5.2.1 Pairings

Let E be an elliptic curve defined over $\mathbb{F}_q$ where $q = p^n$ for some prime p and a positive integer n. For any extension field L of $\mathbb{F}_q$, $E(L)$ denotes the set of L-rational points on E, *i.e.,* the points with coordinates in L, together with the point at infinity ∞. Then $E(L)$ forms a group with identity ∞. Let $\#E(L)$ be the order of this group. Now consider a large prime r dividing $\#E(\mathbb{F}_q)$. Let k be an embedding degree, *i.e.,* the smallest positive integer such that $r \mid q^k - 1$. Consider the r-torsion subgroup $E(\mathbb{F}_{q^k})[r]$. The Tate pairing is a well-defined non-degenerate bilinear map

$$\begin{array}{rcl} \langle \cdot, \cdot \rangle : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) & \to & \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r \\ \big(P, Q + rE(\mathbb{F}_{q^k})\big) & \mapsto & f_{r,P}(D), \end{array}$$

where D is a divisor equivalent to $(Q) - (\infty)$ and $f_{r,P}$ is a function with divisor

$$div(f_{r,P}) = r(P) - (rP) - (r-1)(\infty).$$

Since the image of the pairing is represented by a coset element, to avoid this one can use the reduced Tate pairing

$$e(P,Q) = f_{r,P}(D)^{(q^k-1)/r}.$$

Furthermore, if $(u_\infty f_{r,P})(\infty) = 1$ for some uniformizer u_∞ at ∞, we say that $f_{r,P}$ is normalized. In the case that $f_{r,P}$ is normalized one can simply work with point Q instead of using divisor D

$$e(P,Q) = f_{r,P}(Q)^{(q^k-1)/r}.$$

From now on, we call the function $f_{r,P}$ Miller function and always assume that it is normalized.

Miller algorithm computes Miller function in $\log r$ operations, called Miller length. As in [27, 55], Miller length can be further reduced by defining new pairings based on the Tate pairing. All those variations of the Tate pairing have Miller length at least $\log r/\varphi(k)$. On this line of research, Vercauteren defined the notion of optimal pairings which achieves $\log r/\varphi(k)$ Miller length and proposed an algorithm to obtain a pairing with optimal Miller length for any parametrized pairing-friendly elliptic curve. The notion of pairing-friendly curves will be introduced in the next subsection.

5.2.2 Pairing-Friendly Elliptic Curves

For the security of pairing-based cryptosystems, the discrete logarithm problems (DLP) in the group $E(\mathbb{F}_q)$ and in the multiplicative group $\mathbb{F}_{q^k}^*$ must be infeasible. To avoid DL attack on $E(\mathbb{F}_q)$, r must be sufficiently large where r

is the largest prime dividing $\#E(\mathbb{F}_q)$. And q^k should be chosen large enough so that index calculus attack is infeasible. So k needs to be large enough to avoid index calculus attack but small enough for efficient pairing implementation in extension field arithmetic. Thus in pairing-based cryptography one must find elliptic curves with sufficiently large subgroup of order r and small embedding degree k. We call them pairing-friendly curves. Formal definition is as follows.

Definition 5.2.1 ([18])**.** Suppose that E is an elliptic curve defined over a finite field $\mathbb{F}_q$. E is said to be *pairing-friendly* if

- there is a prime $r \geq \sqrt{q}$ dividing $\#E(\mathbb{F}_q)$, and
- the embedding degree of E with respect to r is less than $(\log r)/8$.

In the construction of pairing-friendly curves, one first finds t, r, q such that there exists an elliptic curve E defined over $\mathbb{F}_q$ that has trace t and a subgroup of order r with prescribed embedding degree k, then uses the complex multiplication method to find an elliptic curve equation.

Definition 5.2.2 ([18])**.** Let $t(x), q(x), r(x)$ be polynomials with rational coefficients where $q(x) = p(x)^n$ for some polynomial $p(x)$ and some positive integer n. If there is an elliptic curve E defined over $\mathbb{F}_{q(x_0)}$ with trace $t(x_0)$ that has a subgroup of order $r(x_0)$ for some integer x_0, then we say that E is a *curve in family* (t, r, q) or (t, r, q) *parameterizes a family of elliptic curves with embedding degree* k. Here $p(x)$ and $r(x)$ represent primes.

In ordinary pairing friendly curves defined over an extension field $\mathbb{F}_{p^n}$ with $n > 1$, result values of the Tate pairing could be contained in a smaller embedding field (for example, $\mathbb{F}_{p^k}$ in the worst case) than expected, *i.e.*, $\mathbb{F}_{p^{nk}}$ [28]. To avoid this potential security loss of the DLP in the embedding field ordinary pairing friendly elliptic curves are preferred to be defined over a prime field $\mathbb{F}_p$. Thus, in the remainder of this paper, we deal with only ordinary elliptic curves defined over a prime field.

5.2.3 Exponentiation Method

The final exponent appearing in the Tate pairing is of the form $(p^k-1)/r$. The exponent splits into

$$(p^k-1)/r = [(p^k-1)/\Phi_k(p)] \cdot [\Phi_k(p)/r],$$

where $\Phi_k(x)$ is the k-th cyclotomic polynomial. By definition of the cyclotomic polynomial

$$(p^k-1)/\Phi_k(p) = \prod_{j|k, j\neq k} \Phi_j(p).$$

Note that $\Phi_j(x)$ is a polynomial in x with coefficients in $\{-1,0,1\}$ for $j < 105$ [29]. Thus raising to the exponent $\Phi_j(p)$ takes only a few Frobenius mapping and some inversions in field arithmetic. Furthermore one can replace an inversion of unitary element by a simple conjugations [49, 48], for example $h = g^{p^{k/2}-1} \in \mathbb{F}_{p^k}$ becomes unitary *i.e.* its norm $N_{\mathbb{F}_{p^k}/\mathbb{F}_{p^{k/2}}}(h) = 1$ for even k. Hence, the exponentiation by $(p^k-1)/\Phi_k(p)$ is relatively easy. In this paper, we focus on the exponentiation by $\Phi_k(p)/r$.

Define $\lambda := \Phi_k(p)/r$ and express λ as base p representation $\lambda = \sum_{i=0}^{\ell-1} \lambda_i p^i$ where $\ell = \lceil \log_p \lambda \rceil$. Then

$$g^\lambda = g^{\lambda_0}(g^p)^{\lambda_1} \cdots (g^{p^{\ell-1}})^{\lambda_{\ell-1}}$$

where the element g to be exponentiated is not a fixed element, but depends on the input P and Q. Note that calculating g^{p^i} can be done easily using Frobenius map when g is an element of a finite field with characteristic p.

When ignoring p-power computation, computing g^λ takes at most $(\log p)$ squarings and $(\log p)$ multiplications in general. Note that $2^\ell - \ell - 1$ multiplications are required to compute $g^{i_0}(g^p)^{i_1}\cdots(g^{p^{\ell-1}})^{i_{\ell-1}}$ where $i_j \in \{0,1\}, j = 0,1,\ldots,\ell-1$ for precomputation. In fact the number of squarings is related to the bit length of λ_i's. More precisely, an exponentiation by λ requires $\max_i(\log \lambda_i)$ squarings. Furthermore, if we use the width w sliding window method, the number of multiplications reduces to $(1/w)\cdot \log p$ with 2^{dw} precomputed elements stored.

If we are working on the family of pairing-friendly curves such as BN curves, then the addition chain method proposed by Scott *et al.* [49] gives an efficient exponentiation method. The method computes $g^x, g^{x^2}, \ldots, g^{x^{\max_i(\deg \lambda_i)}}$ and then exploits the vectorial addition chain to compute the remainder. If the parameter x is chosen to have low Hamming weight, the exponentiation takes only a few multiplications. However the number of squarings still remains $\max_i(\log|\lambda_i|)$. This leads us to a natural question, that is, how further we can reduce the maximum size of λ_i and what the lower bound for this is.

5.3 Reducing the final exponentiation

5.3.1 Polynomial representation of the base-p coefficients

For any given integer λ, the coefficients of λ in the base-p representation have almost same size with the base p on average. In this case, an exponentiation by λ has almost $\log p$ squarings. However for many families of pairing friendly curves the number of squarings is quite smaller than $\log p$.

As an instance, let us consider the final exponentiation step of the BN family of curves [3] which has embedding degree $k = 12$. The final exponent $\lambda(x)$ is equal to $(p(x)^4 - p(x)^2 + 1)/r(x)$. Write $\lambda(x)$ as the base-$p(x)$ representation, say $\lambda(x) = \lambda_0(x) + \lambda_1(x)p(x) + \lambda_2(x)p(x)^2 + \lambda_3(x)p(x)^3$, where

$$\begin{aligned}
\lambda_3(x) &= 1, \\
\lambda_2(x) &= 6x^2 + 1, \\
\lambda_1(x) &= -36x^3 - 18x^2 - 12x + 1, \\
\lambda_0(x) &= -36x^3 - 30x^2 - 18x - 2.
\end{aligned}$$

For the choice of $x = -4647714815446351873$, p is 254-bit and both λ_0 and λ_1 are 192-bit (*i.e.*, $\lambda_0, \lambda_1 \approx p^{192/254}$). Thus the required number of squarings is 192, not 254. Roughly speaking, this comes from the fact that $\lambda_0(x)$ and $\lambda_1(x)$ have small coefficients so that they are close to x^3 rather than x^4 for a large number x.

The above example shows that the polynomial representations of $\lambda(x)$ may give advantages in the final exponentiation step. In this section we examine the polynomial representations of the coefficients and investigate the conditions of the coefficients under which the final exponentiation is efficiently computable.

Through this section, we use notations d_f, $LC(f)$, and $||f||_\infty$ for a polynomial $f(x) = f_0 + f_1x + \cdots + f_nx^n$ which denote the degree of f, the leading coefficient f_n of f, and $\max\{|f_0|, \ldots, |f_n|\}$, respectively. Sometimes we simply write f as a evaluated value of $|f(x)|$ at $x = X$. We also define K_f by $|f_{n-1}| + \cdots + |f_1| + |f_0|$.

As indicated above the size of the value of $f(x)$ at $x = X$ for large X is determined by its degree. The following lemma asserts this.

Lemma 5.3.1. *Suppose $f(x) = f_nx^n + \cdots + f_1x + f_0$, $f_n \neq 0$. For any given $\epsilon > 0$, if $|x| = X$ is large so that $X \geq \frac{K_f}{\epsilon|f_n|} > 1$, then*

$$(1-\epsilon)|f_n|X^n \leq |f(x)| \leq (1+\epsilon)|f_n|X^n.$$

Proof. Let $|f(x)| = |x|^n \cdot \left| f_n + \frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n} \right|$, then by triangle inequality,

$$\begin{aligned} X^n \left(|f_n| - \left| \tfrac{f_{n-1}}{x} + \cdots + \tfrac{f_0}{x^n} \right| \right) &\leq |f(x)| \\ &\leq X^n \left(|f_n| + \left| \tfrac{f_{n-1}}{x} + \cdots + \tfrac{f_0}{x^n} \right| \right). \end{aligned}$$

From the assumption

$$\left| \frac{f_{n-1}}{x} + \cdots + \frac{f_0}{x^n} \right| \leq \frac{|f_{n-1}| + \cdots + |f_0|}{X} = \frac{K}{X} \leq \epsilon \cdot |f_n|.$$

Thus

$$(1-\epsilon)|f_n|X^n \leq |f(x)| \leq (1+\epsilon)|f_n|X^n.$$

□

If the $X = |x|$ is sufficiently large, *i.e.*, ϵ is close to 0, then $|f(x)|$ becomes asymptotically close to $|f_n|X^n$. Thus by the lemma we can regard $|f(X)|$ as $|LC(f)| \cdot |X|^{d_f}$.

Lemma 5.3.2. *Let $(p(x), r(x), t(x))$ be a family of pairing friendly curves with embedding degree k. Let $\varphi := \varphi(k)$ and $\lambda(x) := \Phi_k(p(x))/r(x)$. And let $\lambda_0(x)+\lambda_1(x)p(x)+\cdots \lambda_{\varphi-1}(x)p(x)^{\varphi-1}$ be the base-$p(x)$ representation of $\lambda(x)$. For any given $\epsilon > 0$, choose x so that $|x| = X \geq \max\{\frac{K_p}{\epsilon|LC(p)|}, \frac{K_{\lambda_0}}{\epsilon|LC(\lambda_0)|}, \ldots, \frac{K_{\lambda_{\varphi-1}}}{\epsilon|LC(\lambda_{\varphi-1})|}\}$. If $X^{\alpha_i} \leq |LC(\lambda_i)| \leq X^{\beta_i}$ and $X^{\gamma} \leq |LC(p)| \leq X^{\delta}$ for real $\alpha_i, \beta_i, \gamma$ and δ, then the size of $\max_i |\lambda_i|$, denoted by κ, is bounded as follows,*

$$\begin{aligned} &\tfrac{\max_i\{d_{\lambda_i}+\alpha_i\}\log X - \epsilon_2}{(d_p+\delta)\log X + \epsilon_1} \log p \leq \kappa \\ &\quad \leq \tfrac{\max_i\{d_{\lambda_i}+\beta_i\}\log X + \epsilon_1}{(d_p+\gamma)\log X - \epsilon_2} \log p \end{aligned}$$

where $\epsilon_1 = \log(1+\epsilon)$ and $\epsilon_2 = -\log(1-\epsilon)$.

Proof. By the assumption, for sufficiently large X

$$\begin{aligned} X^{\alpha_i} \leq &\ |LC(\lambda_i)| \leq \ X^{\beta_i}, \\ X^{\gamma} \leq &\ |LC(p)| \leq \ X^{\delta}. \end{aligned}$$

We have, by lemma 5.3.1,

$$\begin{aligned}(1-\epsilon)X^{d_{\lambda_i}+\alpha_i} \leq\ & |\lambda_i(x)| \leq\ (1+\epsilon)X^{d_{\lambda_i}+\beta_i},\\ (1-\epsilon)X^{d_p+\delta} \leq\ & |p(x)| \leq\ (1+\epsilon)X^{d_p+\gamma}.\end{aligned}$$

Thus

$$\tfrac{\left(d_{\lambda_i}+\alpha_i\right)\log X-\epsilon_2}{(d_p+\delta)\log X+\epsilon_1} \leq \tfrac{\log|\lambda_i|}{\log p} \leq \tfrac{\left(d_{\lambda_i}+\beta_i\right)\log X+\epsilon_1}{(d_p+\gamma)\log X-\epsilon_2}.$$

Since $\kappa = \max_i \log|\lambda_i|$, the remain of the proof is obvious. □

Note that if $\alpha_i, \beta_i, \gamma$ and δ are sufficiently small so that $|\lambda_i(x)| \approx X^{d_{\lambda_i}}$ and $|p(x)| \approx X^{d_p}$, then we may assume that $\kappa \approx \frac{\max_i\{d_{\lambda_i}\}}{d_p}\log p$. Thus Lemma 5.3.2 implies that if the coefficients of $\lambda_i(x)$ and $p(x)$ are well-bounded then family accelerates the computation of final exponentiation step. This let us consider a specific class of families of pairing-friendly curves as below.

Definition 5.3.1. Let $(p(x), r(x), t(x))$ be a family of pairing friendly curves. Let k be the embedding degree and $\lambda(x) := \Phi_k(p(x))/r(x)$. Let $\lambda(x) = \lambda_0(x) + \lambda_1(x)p(x) + \cdots \lambda_{\varphi-1}(x)p(x)^{\varphi-1}$ be the polynomial representations of coefficients in the base p. If κ is equal to $\frac{\max_i\{d_{\lambda_i}\}}{d_p}\log p$ then we say that the family is final-exponent friendly (FE-friendly).

We note that in many existing families $\lambda_i(x)$'s have small coefficients, thus can be considered as FE-friendly curves. Before precisely analyzing the final exponentiation-efficiency of polynomial representations, we give an semi p-ary representation * of $\lambda(x)$ in terms of $p(x), r(x)$ and $t(x)$. The expression is useful to have some intuition on in which condition the polynomial representations show the superior final exponentiation-efficiency to numerical representations.

*The word 'semi' means that the given p-ary representation is not exact, since the coefficients in that representation might have larger size than p.

r is prime that divides the order of the elliptic curve group $\#E(\mathbb{F}_p) = p+1-t$ where t is the trace of Frobenius map. Thus we can write $p+1-t = hr$, *i.e.*, $p = hr + (t-1) = hr + u$ for some cofactor h. By Hasse's bound, $|u+1| < 2\sqrt{p}$.

Lemma 5.3.3. *Let $p(x) = h(x)r(x) + u(x)$, then*

$$\begin{aligned}\frac{p(x)^i - u(x)^i}{r(x)} &= h(x)\sum_{j=0}^{i-1} p(x)^j \cdot u(x)^{i-j-1} \\ &= h(x)(p(x)^{i-1} + u(x)p(x)^{i-2} + \cdots + u(x)^{i-1}),\end{aligned}$$

for $i > 1$ and $\frac{p(x)^i - u(x)^i}{r(x)} = h(x)$ for $i = 1$.

Proof. In the proof we abbreviate polynomial $f(x)$ simply to f. The proof uses an induction on i. If $i = 1$ then it is obvious. For $i > 1$, by induction hypothesis,

$$\begin{aligned}\frac{p^{i+1} - u^{i+1}}{r} &= \frac{p(p^i - u^i) + u^i(p-u)}{r} \\ &= p \cdot h(p^{i-1} + up^{i-2} + \cdots + u^{i-1}) + u^i h \\ &= h(p^i + up^{i-1} + \cdots + u^{i-1}p + u^i).\end{aligned}$$

$\square$

Let $f(x)$, $g(x)$ be polynomials with rational coefficients. We denote by $\lfloor f(x)/g(x) \rfloor$ the quotient when $f(x)$ divided by $g(x)$. For example, $\lfloor \frac{ax^2+bx+c}{x} \rfloor = ax + b$. Now we have an alternative expression of polynomial representations.

Lemma 5.3.4. *Let $\lambda(x) := \frac{\Phi_k(p(x))}{r(x)}$, then*

$$\lambda(x) = h(x)\left(p(x)^{\varphi-1} + \sum_{i=1}^{\varphi-1} \left\lfloor \frac{\Phi_k(u(x))}{u(x)^i} \right\rfloor\right) + \frac{\Phi_k(u(x))}{r}.$$

Proof. Let $\Phi_k(x) := x^\varphi + a_{\varphi-1}x^{\varphi-1} + \cdots + a_1x + a_0$, where $\varphi := \varphi(k)$. Simply write $f(x)$ as f.

$$
\begin{aligned}
\frac{\Phi_k(p)}{r} &= \frac{p^\varphi + a_{\varphi-1}p^{\varphi-1} + \cdots + a_1p + a_0}{r} \\
&= \frac{p^\varphi - u^\varphi}{r} + \sum_{i=1}^{\varphi-1} \frac{a_i(p^i - u^i)}{r} + \frac{\Phi_k(u)}{r} \\
&= h\{p^{\varphi-1} + p^{\varphi-2}(u + a_{\varphi-1}) + \\
&\quad p^{\varphi-3}\left(u^2 + a_{\varphi-1}u + a_{\varphi-2}\right) + \cdots\} + \\
&\quad \frac{\Phi_k(u)}{r} \\
&= h\left(p^{\varphi-1} + \sum_{i=1}^{\varphi-1} \lfloor \frac{\Phi_k(u)}{u^i} \rfloor\right) + \frac{\Phi_k(u)}{r}
\end{aligned}
$$

The third equality is followed by Lemma 5.3.3. □

We should note that $\lambda(x)$ in the above lemma is not the perfect base-p representation since the degree of $\lfloor \frac{\Phi_k(u(x))}{u(x)^i} \rfloor$ may exceed or be equal to the degree of $p(x)$ for some i. However, when $\varphi = 2$ or in some specific cases overflow does not happen. Now let us analyze the case $\varphi = 2$, *i.e.*, $k = 3, 4, 6$. Let $\Phi_k(x) = x^2 + ax + b$, where $a, b \in \{0, \pm 1\}$. From Lemma 5.3.4, we see that

$$
\begin{aligned}
\Phi_k(p(x))/r(x) = \;& h(x)p(x) + \{h(x)(u(x) + a) \\
& +(u(x)^2 + au(x) + b)/r(x)\}.
\end{aligned}
$$

Note that $d_u < d_r \le d_p$ and

$$
\begin{aligned}
&\deg\{h(x)(u(x) + a) + (u(x)^2 + au(x) + b)/r(x)\} \\
&= \max\{d_h + d_u, 2d_u - d_r\} \\
&= d_h + d_u \\
&= (d_p - d_r) + d_u \\
&\le d_p - 1,
\end{aligned}
$$

where the second equality comes from

$$(d_h + d_u) - (2d_u - d_r) = d_h + d_r - d_u = d_p - d_u \geq 0.$$

Thus if we let $\lambda_1(x)p(x)+\lambda_0(x)$ be the base-p representation of $\Phi_k(p(x))/r(x)$, then $\lambda_1(x) = h(x)$ and $\lambda_0(x) = h(x)(u(x) + a) + (u(x)^2 + au(x) + b)/r(x)$. So, families of the embedding degree k with $\varphi(k) = 2$ yields the efficient final exponentiation step if $LC(h)$ and $LC(hu) = LC(h)LC(u)$ are both small.

For a larger $\varphi(k)$, it seems hard to control $LC(\lambda_i)$'s because of huge coefficients explosion and frequent overflows occurring in the computation of $\Phi_k(u(x))/(u(x)^i)$'s and $\Phi_k(u(x))/r(x)$ of Lemma 5.3.4. However, one can expect that if $\varphi(k)$, $||q||_\infty$, $||r||_\infty$, and $||u||_\infty$ are small enough, so $LC(\lambda_i)$'s are.

Now we are in a position to describe the lower bound of the number of squarings in the final exponentiation for the polynomial representations.

Theorem 5.3.1. *Suppose $(p(x), r(x), t(x))$ is a family of FE-friendly curves. Let $\rho := d_p/d_r$. If $\max\{d_{\lambda_i} : i = 0, 1, \ldots, \varphi - 1\} \geq d_p - \frac{d_r}{\varphi}$, then*

$$\kappa \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p(x).$$

Proof. By Definition 5.3.1,

$$\kappa = \frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p \geq \frac{d_p - \frac{d_r}{\varphi}}{d_p} \log p \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p.$$

□

At first sight the bound in Theorem 5.3.1 may look unnatural. However, this bound is captured in most cases. More precisely, with high probability $\max_i\{d_{\lambda_i}\} = d_p - 1$ in most cases. And all known methods to construct the family of pairing friendly curves use an irreducible polynomial $r(x)$ to define the extension field $L := \mathbb{Q}[x]/(r(x))$ in order that it contains $\mathbb{Q}(\zeta_k)$ with k-th primitive root of unity ζ_k. Thus $\varphi(k)$ divides d_r. Then,

$$\begin{aligned}
\kappa &= \tfrac{d_p-1}{d_p}\log p \\
&= \left(1-\tfrac{1}{d_p}\right)\log p \\
&= \left(1-\tfrac{1}{\rho d_r}\right)\log p \\
&\geq \left(1-\tfrac{1}{\rho\varphi}\right)\log p.
\end{aligned}$$

In addition we show that $\max_i\{d_{\lambda_i}\} < d_p - \frac{d_r}{\varphi}$ is impossible in the next section. Thus for any family of FE-friendly curves κ is always bounded below.

Example 1

Consider the BN family of curves again. BN curve has $k = 12$ and $\rho = 1$. Then κ is expected to be $\left(1-\frac{1}{\varphi(12)}\right)\log p = (3/4)\log p$. In fact as seen in the beginning of this section, the required squarings are $192 \approx \frac{3}{4}\cdot 254$.

Example 2

Consider the cyclotomic family of curves given by [18] (Construction 6.2) with odd embedding degree k.

$$\begin{aligned}
r(x) &= \Phi_{4k}(x), \\
t(x) &= -x^2+1, \\
p(x) &= \tfrac{1}{4}\left(x^{2k+4}+2x^{2k+2}+x^{2k}+x^4-2x^2+1\right).
\end{aligned}$$

This curve has $\rho = \frac{\deg(p)}{\deg(r)} = \frac{2k+4}{2\varphi(k)} = \frac{k+2}{\varphi(k)}$.

Let us compute $\Phi_k(p)/r$ using Lemma 5.3.4. Since $u(x) = t(x)-1 = -x^2$ and $\Phi_{4k}(x) = \Phi_k(-x^2)$, we have $\Phi_k(u)/r = \Phi_k(-x^2)/\Phi_{4k}(x) = 1$ and

$$\begin{aligned}
\max_i\{d_{\lambda_i}\} &= d_h + (\varphi(k)-1)d_u \\
&= (2k+4-\varphi(4k)) + (\varphi(k)-1)\cdot 2 \\
&= 2k+2 < 2k+4 = d_p.
\end{aligned}$$

In this case, $\max_i\{d_{\lambda_i}\} = d_p - 2$. We expect κ to be $\frac{\max_i\{d_{\lambda_i}\}}{d_p} \log p = \frac{k+1}{k+2} \log p$, and this value is correspond to $\left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$. Thus this family of parameterized curves also already attains the minimum value for κ although it looks seemingly arbitrary.

5.3.2 Reducing the size of base p coefficients

In this section, we propose a general method to reduce the number of squarings in computing the final exponentiation by $\lambda := \Phi_k(p)/r$ for any pairing-friendly curves no matter whether they belong to the parameterized family or not. The main idea is to reduce the maximum size of the coefficients of base p representation of λ since it is closely related to the number of squarings. Since the pairing $e(P,Q)^m$ also defines a non-degenerate bilinear pairing map with m relatively prime to r, we use the exponent $m\lambda$ instead of λ. Using lattice basis reduction algorithm one can find $m\lambda$ whose coefficients in base p representation are small. Throughout this section p, r, t are integers not polynomials.

Observations

Since the reduced Tate pairing is non-degenerate, the map $\bar{e}$ also defines non-degenerate bilinear pairing

$$\bar{e}(P,Q) = e(P,Q)^m = f_{r,P}(Q)^{m(p^k-1)/r},$$

if $\gcd(r,m) = 1$. Let $g := f_{r,P}(Q)^{(p^k-1)/\Phi_k(p)}$, then $\bar{e}(P,Q) = g^{m\lambda}$. We want to find $m\lambda$ with $\gcd(r,m) = 1$ such that

$$m\lambda = \sum_{i=0}^{d-1} v_i p^i,$$

where v_i's are as small as possible. (The choice of d will be given later.) With abuse of notations, we write $\sum_{i=0}^{d-1} v_i p^i = (v_0, v_1, \ldots, v_{d-1})$.

Reducing the coefficients of base p representation

Motivated by [55], $m\lambda$ with small coefficients in base p representation can be obtained by using lattice basis reduction algorithm. Let L be the lattice of dimension d spanned by rows of the matrix

$$\begin{pmatrix} \lambda & 0 & 0 & \cdots & 0 \\ -p & 1 & 0 & \cdots & 0 \\ -p^2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & \\ -p^{d-1} & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

It is easily verified that $v := (v_0, v_1, \cdots, v_{d-1}) \in L$ if and only if $\sum_{i=0}^{d-1} v_i p^i = m\lambda$ for some integer m. Now finding $m\lambda$ with small coefficients reduces to find a short vector in lattice L. By Minkowski's theorem, there is a shortest vector v in L satisfies $||v||_\infty \leq |vol(L)|^{1/d}$ where $||v||_\infty = \max\{|v_i| : i = 0, 1, \ldots, d-1\}$ and $vol(L)$ denotes the volume of L. Then there exists $m\lambda = \sum_{i=0}^{d-1} v_i p^i$ with

$$\begin{aligned} \max\{|v_i|\} &\leq |\det(L)|^{1/d} = |\lambda|^{1/d} \\ &= \left(\tfrac{\Phi_k(p)}{r}\right)^{1/d} \approx (p^{\varphi(k)-1/\rho})^{1/d}. \end{aligned}$$

Since $\Phi_k(p) \equiv 0 \bmod \lambda$, any powers p^i for $i \geq \varphi(k)$ can be represented by a linear combination of $1, p, \ldots, p^{\varphi(k)-1}$ modulo λ and since $\Phi_k(p) = r\lambda$ has small coefficients in base p representation to avoid degenerate pairing maps, it suffices to consider the lattice with dimension $d = \varphi(k)$. Thus κ reduces to $[(\rho \cdot \varphi(k) - 1)/d\rho] \log p = \left(1 - \frac{1}{\rho\varphi(k)}\right) \log p$. And LLL basis reduction algorithm finds a short vector in a low dimensional lattice L.

m is relatively prime to r

$m\lambda$ with small coefficients in base p representation can be obtained efficiently using LLL algorithm. For non-degeneracy of the pairing m must be relatively prime to r. This is equivalent that m is not a multiple of r since r is prime. The following lemma asserts this property.

Lemma 5.3.5. *Let $\lambda := \Phi_k(p)/r$ and $\varphi := \varphi(k)$. Suppose that r is a prime larger than $2^{\varphi(\varphi+1)}$ and p is a prime larger than 3. If $m\lambda = \sum_{i=0}^{\varphi-1} v_i p^i$ with $|v_i| \le \lambda^{1/\varphi}$ and assume that $m = n \cdot r$ for some integer n, then n must be 0.*

Proof. We will use the inequality $(p-1)^\varphi \le \Phi_k(p) \le (p+1)^\varphi$ for all k. The inequality follows from $|\zeta| = 1$ for k-th primitive root of unity ζ in $\mathbb{C}$, $\Phi_k(x) = \prod_{(j,k)=1}(x-\zeta^j)$, the triangular inequality $|x|-1 \le |x-\zeta^j| \le |x|+1$ and $\varphi(k) = \deg(\Phi_k(x))$. First observe that

$$\left(\frac{p}{p-1}\right)^\varphi \cdot \left(\frac{p+1}{p-1}\right) \le 2^{\varphi+1} < r^{1/\varphi}$$

from $p/(p-1) < (p+1)/(p-1) \le 2$ and $r > 2^{\varphi(\varphi+1)}$. From this

$$\frac{p+1}{r^{1/\varphi}} \cdot \frac{p^\varphi}{p-1} < (p-1)^\varphi.$$

Then

$$\begin{aligned}
|n|\Phi_k(p) &= |m\lambda| = |\textstyle\sum_{i=0}^{\varphi-1} v_i p^i| \\
&\le \textstyle\sum_{i=0}^{\varphi-1} \lambda^{1/\varphi} p^i < \lambda^{1/\varphi} \cdot \frac{p^\varphi}{p-1} \\
&\le \left(\frac{(p+1)^\varphi}{r}\right)^{1/\varphi} \cdot \frac{p^\varphi}{p-1} \\
&= \frac{p+1}{r^{1/\varphi}} \cdot \frac{p^\varphi}{p-1} \\
&< (p-1)^\varphi \le \Phi_k(p).
\end{aligned}$$

Hence $|n|\Phi_k(p) < \Phi_k(p)$ and n must be 0. □

In the pairing based cryptosystems, for the 80-bit security r is usually chosen to be 160 bits prime. In this case, if $d = \varphi(k) \le 12$ then r is always larger than $2^{d(d+1)}$. Thus the assumption in lemma holds.

The lower bound for κ

We can reduce the $\kappa(m\lambda)$ to Minkowski's bound, that is $\frac{1}{\varphi(k)}\log\left(\frac{\Phi_k(p)}{r}\right)$ for any pairing-friendly curves by finding a shortest vector in a lattice L. Next, Lemma 5.3.7 shows that $\kappa(m\lambda)$ is bounded below by $\frac{1}{\varphi(k)}\log\left(\frac{\Phi_k(p)}{r}\right) - \log\varphi(k)$.

Lemma 5.3.6. *[34, Theorem 4.4.1] Let $k \geq 2$ and t be positive integers, and let $s = \sum_i s_i t^i$ with $s_i \in \mathbb{Z}$. If $s(x) = \sum_i s_i x^i \not\equiv 0 \bmod \Phi_k(x)$, then*

$$\sum_i |s_i| \geq |\gcd(s, \Phi_k(t))|^{1/\varphi(k)}.$$

Lemma 5.3.7. *Let $m\lambda := m\frac{\Phi_k(p)}{r} = \sum_{i=0}^{\varphi(k)-1}\lambda_i p^i$, where m is coprime to r. Then*

$$||m\lambda||_\infty \geq \frac{1}{\varphi(k)}\left(\frac{\Phi_k(p)}{r}\right)^{1/\varphi(k)}.$$

Proof. Since $\sum_{i=0}^{\varphi(k)-1}\lambda_i x^i \not\equiv 0 \bmod \Phi_k(x)$, by [34, Theorem 4.4.1], we have

$$\begin{aligned} \varphi(k)\cdot||m\lambda||_\infty &\geq \textstyle\sum_i |\lambda_i| \\ &\geq |\gcd(m\lambda, \Phi_k(p))|^{1/\varphi(k)} \\ &= \left(\tfrac{\Phi_k(p)}{r}\right)^{1/\varphi(k)}. \end{aligned}$$

□

Thus for any pairing-friendly curves $\kappa(m\lambda)$ is lower-bounded by $\log\left(\frac{\Phi_k(p)}{r}\right)^{1/\varphi} - \log\varphi$, where m runs through all the integers relatively prime to r. By applying this to the FE-friendly curves, we show that Theorem 5.3.1 is always true without the conditions on degree.

Theorem 5.3.2. *Let $(p(x), r(x), t(x))$ be a family of FE-friendly curves with embedding degree k and let $m(x)\lambda(x) := m(x)\frac{\Phi_k(p(x))}{r(x)} = \sum_i \lambda_i(x)p^i$. For* *given* $0 < \epsilon < 1$, *choose* X *so that* $p = p(X) \geq \frac{K_{\Phi_k}}{\epsilon}$ *and suppose that* $\varphi(k) < (1-\epsilon)\, p^{1/d_p}$. *Then there exists* $i \in \{0, 1, \ldots, \varphi - 1\}$ *such that* $d_{\lambda_i} \geq d_p - d_r/\varphi$ *for any* $m(x)$ *coprime to* $r(x)$.

Proof. Let $\varphi := \varphi(k)$. At first, by Lemma 5.3.1, we have

$$(1-\epsilon)p^{\varphi - 1/\rho} \leq \frac{\Phi_k(p)}{r} \leq (1+\epsilon)p^{\varphi - 1/\rho},$$

where $\rho = \log p / \log r$. By taking the logarithm in the first inequality and dividing by φ, we get

$$\left(1 - \frac{1}{\rho\varphi}\right)\log p - \frac{1}{\varphi}\log\left(\frac{\Phi_k(p)}{r}\right) \leq -\frac{1}{\varphi}\log(1-\epsilon).$$

Now suppose that there exists a curve with $d_{\lambda_i} < d_p - d_r/\varphi$ for all i. Since $\varphi(k)$ divides d_r (see the paragraph below Theorem 5.3.1), the inequalities equivalent to $d_{\lambda_i} \leq d_p - d_r/\varphi - 1$ for all i. If we evaluate $p(x), r(x)$ at $x = X$, then for all i

$$\begin{aligned}
\frac{d_{\lambda_i}}{d_p}\log p &\leq \left(1 - \frac{1}{\rho\varphi}\right)\log p - \frac{1}{d_p}\log p \\
&\leq \frac{1}{\varphi}\log\left(\frac{\Phi_k(p)}{r}\right) - \frac{1}{\varphi}\log(1-\epsilon) - \frac{1}{d_p}\log p \\
&< \frac{1}{\varphi}\log\left(\frac{\Phi_k(p)}{r}\right) - \log\varphi.
\end{aligned}$$

saying that $\kappa(m\lambda) = \frac{\max_i\{d_{\lambda_i}\}}{d_p}\log p < \frac{1}{\varphi}\log\left(\frac{\Phi_k(p)}{r}\right) - \log\varphi$. The last inequality comes from

$$\varphi < (1-\epsilon)\, p^{1/d_p} < (1-\epsilon)^{1/\varphi}\, p^{1/d_p}.$$

However, by Lemma 5.3.7, we must have

$$\kappa(m\lambda) \geq \frac{1}{\varphi}\log\left(\frac{\Phi_k(p)}{r}\right) - \log\varphi,$$

which leads us to a contradiction. $\square$

Since embedding degree k is usually small so that $\varphi(k) < (1-\epsilon)\, p^{1/d_p}$ for a large number p, the assumption in the above theorem holds in most cases. Therefore if $(p(x), r(x), t(x))$ is a family of FE-friendly curves, by taking $m(x) = 1$, then $\kappa(\lambda)$ has a lower bound,

$$\kappa(\lambda) \geq \left(1 - \frac{1}{\rho\varphi}\right) \log p.$$

We note that many existing parameterized families of pairing-friendly curves already attain the prescribed lower bound without modifying λ by a multiple of λ. In these cases, the idea that uses a multiple of λ gives a little advantages for the final exponentiation. See example 1 and 2 in Section III and example 6 in Section V.

5.3.3 Examples

In this section we give some examples investigated by lattice basis reduction. All results satisfy the Minkowski's bounds well as we have shown that theoretically. Our approach using lattice reduction reduces the number of squarings nicely for the curves which are not in the family.

First and second example show the case when our method is applied to DEM curves and third example gives an example applied to Cocks-Pinch curve. Both DEM curve and Cocks-Pinch curve are the curves not in the family.

Example 3

Dupont, Enge, and Morain proposed some parameters for pairing-friendly curves in [17]. The following p and r parameterize the pairing-friendly curve for $k = 5$:

$$\begin{aligned} p &= 91600022435668881297760819108273609 \\ &\quad (117 \text{ bits}), \\ r &= 1040375393410195481 \, (60 \text{ bits}). \end{aligned}$$

Then the final exponent is of the form $\lambda = (p^4 + p^3 + p^2 + p + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$ where

$$\begin{aligned} a_0 &= 48298402242066861357969209793319103 \\ &\quad (116 \text{ bits}), \\ a_1 &= 68283809547505356824804028665198693 \\ &\quad (116 \text{ bits}), \\ a_2 &= 53294610661059016732355697881722241 \\ &\quad (116 \text{ bits}), \\ a_3 &= 88045164289610560 \, (57 \text{ bits}). \end{aligned}$$

Note that the maximum bit length of a_0, a_1, a_2, a_3 is 116 bits. The naive implementation takes totally 115 squarings and 118 multiplications. However, our method finds $m\lambda = b_0 + b_1 p + b_2 p^2 + b_3 p^3$ where

$$\begin{aligned} b_0 &= -2868147363431539633026293965700 \\ &\quad (102 \text{ bits}), \\ b_1 &= -179610012117759028207462943 \, (88 \text{ bits}), \\ b_2 &= 89797974551946435080337006 \, (87 \text{ bits}), \\ b_3 &= 14058171382122118208099 \, (74 \text{ bits}), \\ m &= 159670. \end{aligned}$$

The implementation requires total 101 squarings and 96 multiplications. Consequently our method reduces the number of squarings by 12% and the number of multiplications by 18.6%.

Example 4

Another example in [17] proposes parameters of the curves for $k = 10$:

$$
\begin{aligned}
p &= 265838773006906750756458394131391985 \\
&\quad\; 334144469091740860612401985800108057 \\
&\quad\; 326350300019063611949402010036257572 \\
&\quad\; 717554080849369\,(407 \text{ bits}), \\
r &= 256214560650754227295112990192149027 \\
&\quad\; 295425919988923934988589 41\,(204 \text{ bits})
\end{aligned}
$$

where $\lambda = (p^4 - p^3 + p^2 - p + 1)/r$. The naive implementation requires 405 squarings and 367 multiplications. When our method is applied to the λ, computing the final exponent needs 354 squarings and 339 multiplications with

$$
\begin{aligned}
m &= 673788733967432961409894776 56 \\
&\quad\; 14834417013174705.
\end{aligned}
$$

Thus our method reduces the number of squarings by 12.5% and the number of multiplications by 7.6%.

Example 5

We apply our method to the example of Cocks-Pinch method from p. 211 of [20] for k=12.

$$
\begin{aligned}
p &= 4436167653364218931891\,(72 \text{ bits}), \\
r &= 2147483713\,(32 \text{ bits}).
\end{aligned}
$$

In this case, $\lambda = (p^4 - p^2 + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$ and a_2 have 71 bits. The reduction shows that the maximum bit length of $m\lambda$ is 64 bits with $m = 73639$, so reduces the number of squarings by 9.86%. Next example shows the case when the lattice basis reduction is applied to the families of curves.

Example 6

Consider the BN curves with $x = -4647714815446351873$. These parameters are originally suggested by Nogami *et al.* [44].

$$
\begin{aligned}
p &= 1679810873101583228494080414223173390988918712143906984893371542607275 \\
&\quad 3864723\,(254 \text{ bits}), \\
r &= 1679810873101583228494080414223173390975957960340475274902837886416557 \\
&\quad 0215949\,(254 \text{ bits}).
\end{aligned}
$$

Let $\lambda = (p^4 - p^2 + 1)/r = a_0 + a_1 p + a_2 p^2 + a_3 p^3$, then a_0 and a_1 have 192 bits. So the number of squarings is 191. After the lattice basis reduction we get $m\lambda = b_0 + b_1 p + b_2 p^2 + b_3 p^3$ where b_0 and b_2 have 190 bits with

$$m = 1296075180343170998867457026453982412 83.$$

As we have noted in previous section, BN curves already attain Minkowski's bound. The example shows that there is no noticeable difference by lattice reduction for FE-friendly curves such as BLS curves [2], KSS curves [32].

Chapter 6

Conclusion

In the thesis, we studied on the discrete logarithm problem with auxiliary inputs. By analyzing the non-uniform birthday problem, we reduced the DLPwAI into finding a polynomial with the small value set or whose substitution polynomial has many absolutely irreducible factors as possible. As an rigorous example, we found examples, $f(x) = x^d$ and the Dickson polynomial of degree d. The complexity when it applied to these polynomials coincides with Cheon's algorithm.

If we relax the condition on the degree of the polynomial, it is relatively easy to find such polynomial. With the polynomial, we could solve the generalized DLPwAI efficiently. It would be also interesting to reduce the DLPwAI into the generalized DLPwAI.

As an independent of interest, we described the value set of the generalized Dickson polynomial. It is also of interest to apply this polynomial to solve the DLPwAI.

Finally, we tried to solve the pairing inversion problem which can be used to solve the DLP efficiently. We focused on inverting the final exponentiation step by reducing the final exponentiation. We proposed an universal method to reduce $\kappa(\lambda)$ to $\left(1-\frac{1}{\rho\phi(k)}\right)\log p$, and showed that it is the lower bound for κ. It seems to give another evidence of the hardness of the pairing inversion problem.

Bibliography

[1] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. *IACR Cryptology ePrint Archive*, 2013. `http://eprint.iacr.org/2013/400`.

[2] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. Constructing elliptic curves with prescribed embedding degrees. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 257–267. Springer, 2002.

[3] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2006.

[4] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[5] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptol-*

ogy - EUROCRYPT 2004, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[6] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[7] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2005.

[8] Daniel R. L. Brown and Robert P. Gallant. The static Diffie-Hellman problem. *IACR Cryptology ePrint Archive*, 2004. `http://eprint.iacr.org/2004/306`.

[9] L. Carlitz, D. J. Lewis, W. H. Mills, and E. G. Strauss. Polynomials over finite fields with minimal value sets. *Mathematika*, 8:121–130, 1961.

[10] Seunghwan Chang, Hoon Hong, Eunjeong Lee, and Hyang-Sook Lee. Reducing pairing inversion to exponentiation inversion using non-degenerate auxiliary pairing. *IACR Cryptology ePrint Archive*, page 313, 2013. `http://eprint.iacr.org/2013/313`, to appear Pairing 2013.

[11] Jung Hee Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2006.

[12] Jung Hee Cheon. Discrete logarithm problems with auxiliary inputs. *J. Cryptology*, 23(3):457–476, 2010.

[13] Jung Hee Cheon, Taechan Kim, and Yong Soo Song. The discrete logarithm problem with auxiliary inputs.

[14] Jung Hee Cheon, Taechan Kim, and Yong Soo Song. A Group action on $\mathbb{F}_p{}^{\times}$ and the generalized dlp with auxiliary inputs. In *Selected Areas in Cryptography 2013*.

[15] W.-S. Chou., J. Gomez-Calderon, and G. L. Mullen. Value sets of dickson polynomials over finite fields. *Journal of Number Theory*, 30:334 –344, 1988.

[16] C. Cocks and R. G. E. Pinch. Identity-based cryptosystems based on the Weil pairing. *Unpublished manuscript*, 2001.

[17] Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small mov degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, 2005.

[18] David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2), 2010.

[19] Laura Fuentes-Castañeda, Edward Knapp, and Francisco Rodríguez-Henríquez. Faster hashing to $\mathbf{G}_2$. In *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 412–430. Springer, 2011.

[20] Steven D. Galbraith. *Chapter IX. Pairings in Advances in elliptic curve cryptography*, volume 317 of *London Mathematics Society Lecture Note Series*. Cambridge University Press, 2005.

[21] Steven D. Galbraith. *Mathematics of Public Key Cryptography.* Cambridge University Press, 2012.

[22] Steven D. Galbraith, Florian Hess, and Frederik Vercauteren. Aspects of pairing inversion. *IEEE Transactions on Information Theory*, 54(12):5719–5728, 2008.

[23] Steven D. Galbraith and Mark Holmes. A non-uniform birthday problem with applications to discrete logarithms. *Discrete Applied Mathematics*, 160(10-11):1547–1560, 2012.

[24] Faruk Göloglu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities: Application to discrete logarithms in $\mathbf{F}_{2^{1971}}$.

[25] Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. Solving a 6120-bit dlp on a desktop computer.

[26] J. Gomez-Calderon and D. J. Madden. Polynomials with small value set over finite fields. *Journal of Number Theory*, 28:167–188, 1988.

[27] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[28] Laura Hitt. On the minimal embedding field. In *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 294–301. Springer, 2007.

[29] I. M. Isaacs. *Algebra: A Graduate Course*. Ameriacan Mathematical Society, 2009.

[30] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *Algorithmic Number Theory, 4th International Symposium*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[31] Antoine Joux. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. *IACR Cryptology ePrint Archive*, 2013. `http://eprint.iacr.org/2013/095`.

[32] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 2008.

[33] Naoki Kanayama and Eiji Okamoto. Approach to pairing inversions without solving miller inversion. *IEEE Transactions on Information Theory*, 58(2):1248–1253, 2012.

[34] Minkyu Kim. Integer factorization and discrete logarithm with additional information. *PhD dissertation, Seoul National University*, 2011.

[35] Minkyu Kim, Jung Hee Cheon, and In-Sok Lee. Analysis on a generalized algorithm for the strong discrete logarithm problem with auxiliary inputs. *Mathematics of Computation*, to appear.

[36] Sungwook Kim and Jung Hee Cheon. Fixed argument pairing inversion on elliptic curves. *IACR Cryptology ePrint Archive*, page 657, 2012. `http://eprint.iacr.org/2012/657`.

[37] Taechan Kim and Jung Hee Cheon. A new approach to discrete logarithm problem with auxiliary inputs. *IACR Cryptology ePrint Archive*, 2012. `http://eprint.iacr.org/2012/609`.

[38] Taechan Kim, Sungwook Kim, and Jung Hee Cheon. On the final exponentiation in tate pairing computations. *IEEE Transactions on Information Theory*, 59(6):4033–4041, 2013.

[39] Ueli M. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete algorithms. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 271–281. Springer, 1994.

[40] Ueli M. Maurer and Stefan Wolf. The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.

[41] Victor S. Miller. The weil pairing, and its efficient calculation. *J. Cryptology*, 17(4):235–261, 2004.

[42] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A new traitor tracing. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E85-A(2):481–484, 2002.

[43] Kazuo Nishimura and Masaaki Sibuya. Occupancy with two types of balls. *Annals of the Institute of Statistical Mathematics*, 40(1):77–91, 1988.

[44] Yasuyuki Nogami, Masataka Akane, Yumi Sakemi, Hidehiro Katou, and Yoshitaka Morikawa. Integer variable χ-based ate pairing. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 178–191. Springer, 2008.

[45] J. M. Pollard. Monte carlo methods for index computation $(\bmod p)$. *Mathematics of Computation*, 32(143):pp. 918–924, 1978.

[46] Takakazu Satoh. On pairing inversion problems. In *Pairing*, volume 4575 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2007.

[47] Takakazu Satoh. On generalization of Cheon's algorithm. *IACR Cryptology ePrint Archive*, 2009. `http://eprint.iacr.org/2009/058`.

[48] Michael Scott and Paulo S. L. M. Barreto. Compressed pairings. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 140–156. Springer, 2004.

[49] Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In *Pairing-Based Cryptography - Pairing 2009*, volume 5671 of *Lecture Notes in Computer Science*, pages 78–88. Springer, 2009.

[50] B. I. Selivanov. On waiting time in the scheme of random allocation of coloured particies. *Discrete. Math. Appl.*, 5(1):73–82, 1955.

[51] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[52] Edlyn Teske. On random walks for pollard's rho method. *Mathematics of Computation*, 70:809–825, 2000.

[53] Saburo Uchiyama. Note on the mean value of $v(f)$. *Proc. Japan Acad*, 31:199–201, 1955.

[54] Frederik Vercauteren. The hidden root problem. In *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 89–99. Springer, 2008.

[55] Frederik Vercauteren. Optimal pairings. *IEEE Transactions on Information Theory*, 56(1):455–461, 2010.

[56] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2003.

[57] Changan Zhao, Fangguo Zhang, and Jiwu Huang. A note on the ate pairing. *Int. J. Inf. Sec.*, 7(6):379–382, 2008.

국문초록

현대 암호시스템은 수학적 난제의 어려움에 의하여 그 안전성이 보장된다. 예를 들어, 군 $G = \langle g \rangle$에서 g, g^{α}가 주어진 경우, 이산로그 α를 찾는 문제는 대표적인 암호학적 난제이다. 한편, Generic 군 모델에서 이산로그 알고리즘의 복잡도 하한은 $\Omega(p^{1/2})$로 주어지는데 (단, p는 주어진 군의 소수인 위수), 별도의 부가정보를 이용하면 이보다 쉽게 해결할 수 있는 알고리즘이 존재한다 (Cheon의 알고리즘 등). 본 학위논문에서는 부가적인 입력이 주어진 경우 이산로그문제를 푸는 효과적인 알고리즘에 대하여 연구한다. 한편, 페어링 역연산 알고리즘이 이산로그 문제를 해결할 수 있다는 점에 착안하여 페어링 역연산 알고리즘 복잡도 개선에 대하여 연구한다.

주요어휘: 이산로그문제, 페어링 역연산, Cheon의 알고리즘, Dickson 다항식

학번: 2007-20270

감사의 글

6년전 대학원 진학을 결심하였을 당시에도 연구-새로운 것을 찾는 학문적 행위에 대한 막연한 두려움이 있었습니다. 두려움을 피하고 현실에 안주하려고 할 때마다 적절한 자극으로 이를 돌파할 수 있도록 격려해주신 천정희 선생님이 아니었다면 이 학위논문을 마무리할 수 없었을 것입니다. 감사드립니다.

또한 바쁘신 와중에 저의 논문심사에 귀한 시간 할애하여 주신, 김명환 선생님, 이향숙 선생님, 변동호 선생님, 오병권 선생님께도 감사의 말씀드립니다. 학문적으로 많은 도움을 주신 암호랩의 이인석 선생님, 홍진 선생님께도 감사드립니다.

본 학위논문의 이론적인 부분에 많은 도움을 주신 재홍 형, (류)한솔, 용수에게도 감사드립니다. I also would like to appreciate S. Galbraith, I. Shparlinski, M. Tibouchi, and M. Zieve for various discussions.

4년전 처음 암호랩에 들어온 이후로 아낌없는 조언과 고민을 나누었던 선배님들, 명선 형, 민규 형, 성욱 형, 형태 형이 계셨던 덕분에 힘들 때 위로가 되고 기쁠 때 행복할 수 있었습니다. 또한, 비슷한 시기에 (간발의 차이로 늦게) 랩에 들어와 저 대신 고생 많이 했던 후배 진수, 병도와 한 연구실에서 오랜 시간 함께했던 치홍 형, 현숙, 희원, 미란, 민영, 창민, ISaC의 충훈 형, 민재 형, 가원, 병일 여러분 덕분에 좋은 추억 만들 수 있었습니다.

학부 동기로 오랜 시간 함께 한 진영, 지웅, 공, 지훈, (홍)한솔, 대학원 동기로 함께 즐거운 시간 보냈던 성준 형, 호중 형, 성환 형, 민하 형, 경석 형, 정태 형, 문창 형, 철홍 형, 형석 형, 수정 누나, 세진 형 모두 각자의 분야에서 멋지게 활약하기를 기원합니다. 이외에 지면에 미처 담지 못한 수많은 지인들에게 너그러운 마음으로 이해를 구하며 감사의 말씀 전합니다.

긴 세월 동안 묵묵히 믿고 지켜봐주신 양가 부모님과 동생들, 처형 내외 덕분에 꿋꿋하게 힘든 시간 버틸 수 있었습니다. 끝으로 모진 남편, 아빠 때문에 고생많았던 마누라 효민과 아들 휘성에게 지금까지 곁에서 잘 지켜줘서 고맙고 함께할 남은 평생 동안에도 서로 이해하며 슬기롭게 헤쳐나갈 수 있기를 바라는 마음으로 이 학위논문을 바칩니다.