LMS J. Comput. Math. 19 (1) (2016) 1–15 © 2016 Authors
doi:10.1112/S1461157015000303

A new approach to the discrete logarithm problem with auxiliary inputs

Jung Hee Cheon and Taechan Kim

Abstract

The aim of the discrete logarithm problem with auxiliary inputs is to solve for α, given the elements $g, g^{\alpha}, \ldots, g^{\alpha^d}$ of a cyclic group $G = \langle g \rangle$, of prime order p. The best-known algorithm, proposed by Cheon in 2006, solves for α in the case where $d \mid (p \pm 1)$, with a running time of $O(\sqrt{p/d} + d^i)$ group exponentiations ($i = 1$ or $1/2$ depending on the sign). There have been several attempts to generalize this algorithm to the case of $\Phi_k(p)$ where $k \geqslant 3$. However, it has been shown by Kim, Cheon and Lee that a better complexity cannot be achieved than that of the usual square root algorithms.

We propose a new algorithm for solving the DLPwAI. We show that this algorithm has a running time of $\widetilde{O}(\sqrt{p/\tau_f} + d)$ group exponentiations, where τ_f is the number of absolutely irreducible factors of $f(x) - f(y)$. We note that this number is always smaller than $\widetilde{O}(p^{1/2})$.

In addition, we present an analysis of a non-uniform birthday problem.

1. *Introduction*

1.1. *The discrete logarithm problem with auxiliary inputs*

The discrete logarithm problem with auxiliary inputs (DLPwAI) for a group G, of prime order p, can be stated as follows. Given the elements $g, g^{\alpha}, \ldots, g^{\alpha^d} \in G$, compute α. A number of variants of the DLP, such as the Weak Diffie–Hellman Problem (WDHP) [16], the Strong Diffie–Hellman Problem (SDHP) [2], the Bilinear Diffie–Hellman Inversion Problem (BDHIP) [1] and the Bilinear Diffie–Hellman Exponent Problem (BDHEP) [3] ask for the determination of some values encoded by the discrete logarithm α, for some given $g, g^{\alpha}, \ldots, g^{\alpha^d} \in G$. Therefore, solving the DLPwAI implies that these problems are also solved. These problems arise in a number of contexts. For example, traitor tracing [16], short signatures [2], ID-based encryption [1] or broadcast encryption [3].

The state-of-the-art algorithm for this problem was proposed by Cheon [5, 6], and Brown and Gallant [4]. It has a running time of $O(\sqrt{p/d} + \sqrt{d})$ group exponentiations in the case where $d \mid (p-1)$, and $O(\sqrt{p/d} + d)$ in the case where $d \mid (p+1)$. The idea of Cheon's algorithm is to embed the discrete logarithm α into the finite fields $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. He exploits the fact that α^d can be embedded into an element of a small subgroup of $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$, when d is a divisor of $p \pm 1$.

Subsequently, several generalizations of this algorithm have attempted to solve the problem when d is a divisor of $\Phi_k(p)$ for the kth cyclotomic polynomial $\Phi_k(x)$ [7, 14, 20]. Satoh [20] generalized the algorithm, using the embedding of $\alpha \in \mathbb{F}_p$ into the general linear group $\mathrm{GL}_k(\mathbb{F}_p)$. However, its complexity for $k \geqslant 3$ was not well understood. More recently, Kim, Cheon and Lee [14] noticed that Satoh's generalization is essentially the same as the embedding

Received 19 February 2015; revised 4 September 2015.

2010 Mathematics Subject Classification 68Q25 (primary), 11Y16 (secondary).

The first author was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-TB1403-00. A part of the paper appears in the PhD Dissertation of the second author.

of $\mathbb{F}_p$ into $\mathbb{F}_{p^k}$, and clarified the complexity of the algorithm. Unfortunately, their result suggests that the complexity of this generalization is not faster than that of current square root complexity algorithms, such as Pollard's rho algorithm [19] for $k \geqslant 3$.

Cheon *et al.* [8] recently presented an algorithm for solving α when neither $p+1$ nor $p-1$ has an appropriate small divisor d. Specifically, they solve for α when $g^{\alpha^{k_1}}, \ldots, g^{\alpha^{k_d}}$ are given, and the set of elements k_i forms a multiplicative subgroup of $\mathbb{Z}_{p-1}^{\times}$. However, a way to reduce the DLPwAI to this problem is not currently known.

1.2. *Our contributions*

We present a new algorithm for solving the DLPwAI. The proposed algorithm has a running time of less than $\widetilde{O}(p^{1/2})$, in any case where $d < p^{1/2}$. We briefly describe the algorithm, as follows. First, one chooses a non-zero polynomial f, of degree d. Then, one randomly chooses some elements r_i and s_j from $\mathbb{F}_p$, and computes two lists (the value of m will be determined later)

$$L_1 := \{g^{f(r_i\alpha)} : r_i \in \mathbb{F}_p, 1 \leqslant i \leqslant m\} \quad \text{and} \quad L_2 := \{g^{f(s_j)} : s_j \in \mathbb{F}_p, 1 \leqslant j \leqslant m\}.$$

If the two lists have an element in common, say, $f(r_{i_0}\alpha) = f(s_{j_0})$, then finding the roots of $\tilde{f}(\alpha) := f(r_{i_0}\alpha) - f(s_{j_0})$ in $\mathbb{F}_p$ gives d candidates for the desired solution.

In order to refine the complexity, we consider several problems. As a first step, the computation of the list L_1 can be thought of as the problem of computing a multipoint evaluation when the coefficients of a polynomial are exponentiated. That is, the problem of computing $g^{q(r_1)}, \ldots, g^{q(r_m)}$ for given $g^{q_0}, \ldots, g^{q_d}$, where $q(x) := q_0 + q_1 x + \ldots + q_d x^d$. A naive approach would be to take $O(m \cdot d)$ operations in G. However, we can obtain a fast multipoint evaluation method, computing the list within $\widetilde{O}(m+d)$ group operations, by using the usual fast multipoint evaluation method. A similar result was proposed in [17], using fast Fourier transform (FFT) multiplication. We note that the technique can also be extended to the Schönhage–Strassen multiplication algorithm.

If the size of the image of f is N, then the birthday paradox (under the assumption that f is a random function) suggests that the lists L_1 and L_2 yield a collision with a high probability for $m = O(N^{1/2})$. In order to obtain a more precise collision probability, we consider a non-uniform birthday problem. Suppose that there exist N bins, and a randomly sampled ball is assigned to the bin $k \in \{1, 2, \ldots, N\}$ with a probability of w_k. Then, our analysis shows that the probability of a bin containing at least two different balls after r samplings is non-negligible for $r \geqslant 1/\sqrt{\sum_{k=1}^{N} w_k^2}$. Applying this result to our case, we find that the two lists have a collision with a high probability after $O(1/\sqrt{\sum_{k=1}^{N} w_k^2})$ samplings.

The birthday problem of non-uniform distributions has been dealt with in several contexts [9, 18, 21]. Although the expected number of trials until a collision is precisely determined in these cases, their results only apply when the probability w_k is bounded by c/N, for some constant c that is independent of N. We remark that our analysis applies even when w_k is not well bounded, for example, $w_k = N^{-O(1)}$.

Let ρ_f be the number of rational points over $\mathbb{F}_p$, on the curve defined by $f(x) - f(y) = 0$. Then, as in [10, 15], we can see that $\sum_k {w_k}^2 = \rho_f / p^2$. From this, we derive that the overall complexity of the proposed algorithm is given by $\widetilde{O}(\sqrt{p^2/\rho_f} + d)$ group exponentiations. From Weil's theorem, we have that $\rho_f = \tau_f p \pm O(d^2\sqrt{p})$, where τ_f is the number of absolutely irreducible (that is, defined over $\mathbb{F}_p$ and irreducible over its algebraic closure) factors of $f(x) - f(y)$. In order to obtain a better complexity, we need to find a polynomial $f \in \mathbb{F}_p[x]$ with the largest possible number for τ_f.

We show that τ_f is at most $\sum_{D|d}(\varphi(D)/\mathrm{ord}_D(p))$, where $\mathrm{ord}_D(p)$ is the multiplicative order of p modulo D. In particular, in the case of $d \mid \Phi_k(p)$ for the prime k, we have

$\tau_f \leqslant (d - \gcd(d, p-1))/k + \gcd(d, p-1)$. When $d \mid (p-1)$, one has $\tau_f = d$, because the polynomials $f(x) = x^d$ and $f(x) - f(y)$, factorize into all linear factors. In the case where $d \mid (p+1)$, one has $\tau_D = (d - \gcd(d, p-1))/2 + \gcd(d, p-1)$, because the Dickson polynomial $D(x)$, and $D(x) - D(y)$, factorize into all quadratics other than one or two linear exceptions. Applying the proposed algorithm, it takes $\widetilde{O}(\sqrt{p/d} + d)$ group exponentiations to compute the discrete logarithm α. In the case where $d \mid \Phi_3(p) = (p^2+p+1)$, we show that $f(x) - f(y)$ cannot have an absolutely irreducible cubic factor for any polynomial f. Therefore, it is impossible to achieve the upper bound of τ_f in this case.

The rest of the paper is organized as follows. We begin with a description of the algorithm, and present a complexity analysis, in § 2. In § 3, we present a fast multipoint evaluation method on exponents. The analysis of the birthday problem with a non-uniform distribution is presented in § 4. In § 5, we discuss the choice of polynomials that attain the proposed upper bound of τ_f. We summarize our results and suggest directions for future work in § 6.

2. *The main algorithm*

In this section, we present an algorithm for solving the DLPwAI, with a function defined by a polynomial $f \in \mathbb{F}_p[x]$. Throughout the paper, $\Phi_k(x)$ denotes the kth cyclotomic polynomial, and $\varphi(k)$ is the Euler-totient function.

2.1. *Algorithm description*

Let $G = \langle g \rangle$ be a group of prime order p. The aim is to solve for α, given $g, g^\alpha, \ldots, g^{\alpha^d} \in G$. Cheon's algorithm and its generalizations use an embedding of the discrete logarithm $\alpha \in \mathbb{F}_p$ to auxiliary groups, such as extension fields of $\mathbb{F}_p$. However, the recent result of Kim *et al.* [**14**] shows that the complexity of the several generalizations in the case where $d \mid \Phi_k(p)$ for $k \geqslant 3$ [**7**, **20**] is always greater than $p^{1/2}$. Therefore, we need to consider a different approach to solving the DLPwAI.

First, we choose a polynomial $f \in \mathbb{F}_p[x]$, of degree d. The proposed algorithm employs a map defined by the polynomial f. While previous algorithms require algebraic structures of the auxiliary groups, we concentrate solely on the value set of the polynomial f. A brief description of the algorithm is as follows.

Step 1: For given $f \in \mathbb{F}_p[x]$ and $g, g^\alpha, \ldots, g^{\alpha^d} \in G$, compute two lists

$$L_1 := \{g^{f(r_i\alpha)} : r_i \in \mathbb{F}_p, 1 \leqslant i \leqslant m\} \quad \text{and} \quad L_2 := \{g^{f(s_j)} : s_j \in \mathbb{F}_p, 1 \leqslant j \leqslant m\},$$

where r_i and s_j are randomly chosen from $\mathbb{F}_p$, and m is a positive integer to be determined later.

Step 2: Find a non-empty intersection between L_1 and L_2 or a collision of two elements in L_1, if it exists. If not, repeat Step 1.

Step 3: Recover α by finding roots of $\tilde{f}(\alpha) := f(r_{i_0}\alpha) - f(s_{j_0})$ in $\mathbb{F}_p$, and using (g, g^α) to identify α.

We closely examine the complexity of the proposed algorithm in the next subsection.

2.2. *Complexity analysis*

Consider a naive analysis of the algorithm. First, suppose that the value set $V(f) := \{f(x) : x \in \mathbb{F}_p\}$ is of size N. Assume that the map $x \mapsto f(x)$ behaves as a random function. Then, by the birthday paradox, we expect the lists L_1 and L_2 to have an element in common for $m = O(N^{1/2})$, with a high probability. Next, a naive approach to computing L_1 would take $O(md)$ exponentiations in G. Overall, the complexity of the algorithm is found to be at least $\Omega(N^{1/2})$. However, for a random polynomial of degree d, the average size of the value set of f

is about $(1-1/2!+\ldots+(-1)^{d-1}/d!)\cdot p\approx(1-1/e)\cdot p$ [22], where e denotes the base of the natural logarithm. Therefore, the complexity is already greater than $\Omega(p^{1/2})$.

In order to obtain a better complexity for the algorithm, we should consider several problems. The following theorem shows that we can compute L_1 within $\widetilde{O}(m)$ exponentiations in G.

THEOREM 2.1. *Suppose that an algorithm multiplying two polynomials of degree less than d has a running time of $M(d)$ operations in $\mathbb{F}_p$. Let $f(x)=a_0+a_1x+\ldots+a_{d-1}x^{d-1}\in\mathbb{F}_p[x]$. Given $g^{f(x)}:=(g^{a_0},g^{a_1},\ldots,g^{a_{d-1}})$, and random $r_1,\ldots,r_d\in\mathbb{F}_p$, one can compute $g^{f(r_0)},\ldots,g^{f(r_{d-1})}$ within $O(M(d)\log d)$ operations in G.*

Proof. A sketch of the proof will be given in § 3. □

We also note that the map defined by the polynomial is not, in general, a random function. In this case, as opposed to the random case, the values of the map are non-uniformly distributed. Intuitively, one might expect more collisions for value sets in the non-uniform case. This leads us to consider the birthday problem with a non-uniform distribution. The following result is simpler than the problem studied in the papers [9, 18, 21], but it is sufficient for our applications.

THEOREM 2.2. *For a positive integer N, and some $k\in\{1,2,\ldots,N\}$, let w_k be the probability that a randomly sampled ball is put into the bin k. Let S_r be the probability that a collision occurs in r trials. Assume that $W=\max_k\{w_k\}\leqslant\frac{1}{8}$, and let $D=1/(\sum_k w_k^2)$. If $r\geqslant\sqrt{D+\frac{1}{4}}+\frac{1}{2}\geqslant 5$, then $S_r\geqslant\frac{1}{88}$.*

Proof. The proof will be given in § 4. □

Let $V(f):=\{f(x):x\in\mathbb{F}_p\}=\{y_1,\ldots,y_N\}$ be the value set of a polynomial $f\in\mathbb{F}_p[x]$ of degree d. Let $R_i:=|\{y\in V(f):|f^{-1}(y)|=i\}|$ and $\rho_f:=|\{(x,y)\in\mathbb{F}_p\times\mathbb{F}_p:f(x)=f(y)\}|$. Then

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

and we can see that $p\leqslant\rho_f\leqslant dp$.

Using the above theorems, we can obtain our main theorem. It shows that the proposed algorithm has a running time of $\widetilde{O}(p/\sqrt{\rho_f}+d)$ group operations, which is always $\widetilde{O}(p^{1/2})$.

THEOREM 2.3. *Use the notation as described above. Let $f(x)=a_0+a_1x+\ldots+a_dx^d\in\mathbb{F}_p[x]$ be a polynomial of degree d. Let $G=\langle g\rangle$ be a cyclic group of prime order p. Suppose that an algorithm multiplying two polynomials of degree less than d has a running time of $M(d)$ operations in $\mathbb{F}_p$. Let m be a positive integer such that $5\leqslant\sqrt{p^2/\rho_f+1/4}+1/2\leqslant 2m\leqslant C\cdot\sqrt{p^2/\rho_f}$ for some constant C. For a given $f(x)$, and $g^{\alpha},\ldots,g^{\alpha^d}$, one has a probabilistic algorithm that computes α in $O(m/d\cdot M(d)\log d+d)=O(p/\sqrt{\rho_f}\cdot(M(d)\log d)/d+d)$ operations in G, and an expected number of $O(M(d)\log d\log(dp))$ operations in $\mathbb{F}_p$. The success probability of the algorithm is at least $1/172$, which is non-negligible.*

Proof. The algorithm is described as follows.

(i) Given $f(x)=a_0+a_1x+\ldots+a_dx^d$ and $g,g^{\alpha},\ldots,g^{\alpha^d}$, one can compute $g^{a_0},(g^{\alpha})^{a_1},\ldots,(g^{\alpha^d})^{a_d}$ in $(d+1)$ group exponentiations. Given

$$f_\alpha(x):=f(\alpha x)=a_0+(\alpha\cdot a_1)x+(\alpha^2\cdot a_2)x^2+\ldots+(\alpha^d\cdot a_d)x^d,$$

we denote $g^{f_\alpha(x)}=(g^{a_0},(g^{\alpha})^{a_1},\ldots,(g^{\alpha^d})^{a_d})$.

(ii) Choose random elements $r_1, \ldots, r_m \in \mathbb{F}_p$, and compute the list $L_1 = \{g^{f_\alpha(r_1)}, \ldots, g^{f_\alpha(r_m)}\}$ in $\lceil m/d \rceil \cdot O(M(d) \log d)$ group operations in G, by Theorem 2.1.

(iii) Choose random elements $s_1, \ldots, s_m \in \mathbb{F}_p$, and compute $f(s_1), \ldots, f(s_m)$ using the standard fast multipoint evaluation method, in $\lceil m/d \rceil \cdot O(M(d) \log d)$ operations in $\mathbb{F}_p$.

(iv) Raise to the power of $f(s_i)$ for each $i = 1, \ldots, m$, to obtain the list $L_2 = \{g^{f(s_1)}, \ldots, g^{f(s_m)}\}$. This requires m group exponentiations.

(v) Find a collision satisfying $g^{f_\alpha(r_i)} = g^{f(s_j)}$ or $g^{f_\alpha(r_i)} = g^{f_\alpha(r_{i'})}$ for some indices i, i' and j. It happens with the probability at least $1/172$, as we shall see below.

(vi) Compute at most d candidates for α, by finding roots in $\mathbb{F}_p$ of

$$\widetilde{f}_1(\alpha) := f_\alpha(r_i) - f(s_j) \quad \text{or} \quad \widetilde{f}_2(\alpha) := f_\alpha(r_i) - f_\alpha(r_{i'}).$$

It takes an expected number of $O(M(d) \log d \log(dp))$ operations in $\mathbb{F}_p$, using the root finding algorithm [**23**, Corollary 14.16].

(vii) Identify the exact solution α from d candidates exhaustively, using the information of (g, g^α), which takes d group operations.

We assume that $\alpha \neq 0$ otherwise $g^\alpha = 1$ so one can easily deduce the discrete logarithm from the given instances.

Consider the collision probability in Step (v). The probability that a collision occurs can be regarded as a non-uniform birthday problem. We throw a randomly chosen ball, say, $r_i\alpha$ or s_j, and put it into a bin numbered by $f(r_i\alpha)$ or $f(s_j)$. After repeating this experiment several times, we want to compute the probability that a bin contains at least two balls, that is, that a collision occurs.

Let w_k be the probability that a ball is thrown into the bin k. Then each probability is given by (after proper reordering)

$$(w_{y_1}, \ldots, w_{y_N}) = \Bigl(\underbrace{\frac{1}{p}, \ldots, \frac{1}{p}}_{R_1}, \underbrace{\frac{2}{p}, \ldots, \frac{2}{p}}_{R_2}, \ldots, \underbrace{\frac{d}{p}, \ldots, \frac{d}{p}}_{R_d}\Bigr).$$

After computing the lists L_1 and L_2, one has three possible types of collision: (1) $g^{f_\alpha(r_i)} = g^{f_\alpha(r_{i'})}$ for some i and i', (2) $g^{f(s_j)} = g^{f(s_{j'})}$ for some j and j' and (3) $g^{f_\alpha(r_i)} = g^{f(s_j)}$ for some i and j. Among these three types of collision, the second one is useless in the sense that it does not reveal any information about α. Now we consider the probability that only useful collisions occur.

Let $U := \{(u_1, \ldots, u_m, u_{m+1}, \ldots, u_{2m}) \in \mathbb{F}_p^{2m} : \exists(i,j) \text{ such that } i \neq j \text{ and } f(\gamma_i u_i) = f(\gamma_j u_j)\}$, where $\gamma_i := 1$ for $1 \leqslant i \leqslant m$ and $\gamma_i := \alpha$ for $m+1 \leqslant i \leqslant 2m$. Then U is the set of possible choices of $(r_1, \ldots, r_m, s_1, \ldots, s_m)$ that lead to any collision in the above algorithm. Thus, we have $S_{2m} = |U|/p^{2m}$, where S_{2m} denotes the probability of any collision after throwing $2m$ balls.

On the other hand, let

$$\begin{aligned} U_1 := \{(u_1, \ldots, u_{2m}) \in U : \exists(i,j) \text{ such that} \\ [i \neq j \text{ and } f(\gamma_i u_i) = f(\gamma_j u_j) \text{ and } \max\{i,j\} > m]\}, \end{aligned}$$

and

$$U_2 := \{(u_1, \ldots, u_{2m}) \in U : f(\gamma_i u_i) = f(\gamma_j u_j) \text{ for } i \neq j \text{ implies } \max\{i,j\} \leqslant m\}.$$

Then U_1 is the set of $(r_1, \ldots, r_m, s_1, \ldots, s_m)$ that leads an useful collision (types (1) and (3)), and U_2 is the set of the elements that lead to a useless collision (type (2)). Obviously, these two sets are disjoint and $U = U_1 \cup U_2$. Now one has an injection from U_2 to U_1 given by $(u_1, \ldots, u_m, u_{m+1}, \ldots, u_{2m}) \mapsto (u_{m+1}, \ldots, u_{2m}, \alpha^{-1}u_1, \ldots, \alpha^{-1}u_m)$. Thus we deduce that

$|U_2| \leqslant |U_1|$. Furthermore, the probability of a useful collision, required in Step (v), which is equal to $\Pr[U_1]$, satisfies

$$2\Pr[U_1] = \frac{2|U_1|}{p^{2m}} \geqslant \frac{|U_1|+|U_2|}{p^{2m}} = \frac{|U|}{p^{2m}} = S_{2m}.$$

By Theorem 2.2 and the choice of m, $\Pr[U_1] \geqslant S_{2m}/2 \geqslant \frac{1}{2}\cdot\frac{1}{88} = \frac{1}{172}$. Thus the success probability of the algorithm is at least $\frac{1}{172}$, which is non-negligible. □

Remark 1. The multiplication cost $M(d)$ is $O(d\log d)$, when using the FFT method, and $O(d\log d\log\log d)$, when using the Schönhage–Strassen (SS) method. In both cases, the complexity of the proposed algorithm is bounded by $\widetilde{O}(\sqrt{p^2/\rho_f}+d)$ operations in G, without the log factors $\log d\log\log d$ for the FFT method (or $\log^2 d\log\log d$, for the SS method).

In the following sections, we will present the omitted proofs, and discuss several polynomials that are suitable for the proposed algorithm.

3. *Fast multipoint evaluation on exponents*

In this section, we discuss the polynomial evaluation method when the coefficients of a polynomial are exponentiated. That is, the computation of $g^{f(r_1)},\ldots,g^{f(r_d)}$, when $g^{f(x)} := (g^{a_0},\ldots,g^{a_{d-1}})$ is given for a polynomial $f(x) = a_{d-1}x^{d-1}+\ldots+a_1x+a_0 \in \mathbb{F}_p[x]$.

Given $g^{f(x)}$ and $h(x)$, where $f(x)$ and $h(x)$ are polynomials of degree less than d, one can compute $g^{f(x)h(x)}$ and $g^{f(x)+h(x)}$ in $O(d^2)$ and $O(d)$ exponentiations in G, respectively. Furthermore, one can apply a fast multiplication method, such as the FFT method or the SS method, in order to compute $g^{f(x)h(x)}$ in $\widetilde{O}(d)$ exponentiations in G.

From our observations, it is easy to obtain a fast multipoint evaluation on the exponentiated elements. A similar method is used in [17], with the coefficients of the polynomial being encrypted by an additive homomorphic encryption scheme. It was shown there that the evaluation requires $O(M(d)\log d)$ homomorphic operations (additions and scalar multiplications), where $M(d)$ is the computational cost of the FFT multiplication. It follows from that observation that the FFT multiplication algorithm can be analogously applied to compute $\mathrm{Enc}(f\cdot\tilde{f})$, for given $\mathrm{Enc}(f)$ and $\tilde{f}$, in $M(d)$ homomorphic operations. Here, Enc is the additive homomorphic encryption and $\mathrm{Enc}(f) := (\mathrm{Enc}(a_0),\ldots,\mathrm{Enc}(a_{d-1}))$. This technique can also be applied to our case, by simply replacing $\mathrm{Enc}(a_i)$ with g^{a_i}.

However, the FFT multiplication only works when $\mathbb{F}_p$ contains a dth root of unity, that is, $d \mid (p-1)$. In our application, $(p-1)$ does not necessarily have a proper divisor d, so we note that a multipoint evaluation on the exponentiated elements is also possible using the SS multiplication method. In the SS multiplication, the field $\mathbb{F}_p$ can be arbitrary.

We briefly describe the algorithm as follows.

3.1. *Schönhage–Strassen multiplications*

Suppose that $\deg(fh) \leqslant d = 2^k$, for $m = 2^{\lfloor k/2\rfloor}$ and $t = d/m$. Write down the polynomial as $f(x) = A_0(x)+A_1(x)x^m+\ldots+A_{t-1}(x)x^{m(t-1)}$, where $A_i \in \mathbb{F}_p[x]$ with degree less than m, and let $\bar{f}(x,y) := A_0(x)+A_1(x)y+\ldots+A_{t-1}(x)y^{t-1} \in \mathbb{F}_p[x,y]$, so that $\bar{f}(x,x^m) = f(x)$.

Consider the ring $D := \mathbb{F}_p[x]/(x^{2m}+1)$, and let $\zeta := x \bmod (x^{2m}+1) \in D$ be an element corresponding to x in $\mathbb{F}_p[x]/(x^{2m}+1)$. Then, we can regard $f^*(y) := \bar{f}(\zeta,y) = A_0(\zeta)+A_1(\zeta)y+\ldots+A_{t-1}(\zeta)y^{t-1}$ as a polynomial in y, with coefficients in D. For two polynomials f and h, the SS multiplication computes $f^*(y)h^*(y) \bmod y^t+1$, which is equivalent to $f(x)h(x) \bmod x^d+1$.

Since $\zeta^{2m} = -1$, ζ is a $4m$th primitive root of unity in D, $\eta = \zeta^2$ (or $\eta = \zeta$) is a primitive $2t$th root of unity in D, when $t = m$ (or $t = 2m$, respectively). Now computing

$f^*(y)h^*(y) \bmod (y^t+1)$ is equivalent to computing $f^*(\eta y)h^*(\eta y) \bmod (y^t-1)$. This can be done using the FFT method, with the tth primitive root of unity $\omega = \eta^2$ in D. The multiplication in D can be carried out recursively, with polynomials of degree less than $2m$. 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 $g^{A_i} = (g^{a_{mi}}, g^{a_{mi+1}}, \ldots, g^{a_{mi+(m-1)}})$.

Algorithm 1 Schönhage–Strassen Multiplication (in exponential form)

```
Input: $d = 2^k \in \mathbb{N}$, an element $g$ of order $p$, $(g^{a_0}, g^{a_1}, \ldots, g^{a_{d-1}})$ and $(b_0, \ldots, b_{d-1})$, where
    $f(x) = a_0 + \ldots + a_{d-1}x^{d-1}$ and $h(x) = b_0 + \ldots + b_{d-1}x^{d-1}$ with $\deg(fh) < d$.
Output: $g^{f(x)h(x)} := (g^{c_0}, g^{c_1}, \ldots, g^{c_{d-1}}) \in G^d$
  (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 $f(x) = \sum_{i=0}^{t-1} A_i(x)x^{mi}$,
      $h(x) = \sum_{i=0}^{t-1} B_i(x)x^{mi}$, where $A_i, B_j \in \mathbb{F}_p[x]$ of degree less than $m$.
  (2) Let $D = \mathbb{F}_p[x]/(x^{2m}+1)$ and $\zeta \leftarrow x \bmod (x^{2m}+1)$.
      If $t = 2m$, then $\eta \leftarrow \zeta$. Otherwise, $\eta \leftarrow \zeta^2$ ($\eta$ is a primitive $2t$th root of unity).
      Compute $g^{c^*(\eta y)} = g^{f^*(\eta y)h^*(\eta y) \bmod (y^t-1)}$ with a $t$th root of unity $\eta^2$ using the FFT
      method as described in [17].
      Call the algorithm 1 recursively to compute multiplications in $D$.
  (3) Return $g^{c^*(y)} = (g^{C_0}, \ldots, g^{C_{t-1}})$.
```

Proof of Theorem 2.1. The analysis of the complexity easily follows by replacing the addition/multiplication in the field $\mathbb{F}_p$ with the multiplication/exponentiation in the group G. In the case where the FFT multiplication is used, we refer to [17]. The original SS multiplication takes $O(d \log d \log\log d)$ operations in $\mathbb{F}_p$, so the SS multiplication in the exponential form requires $O(d \log d \log\log d)$ operations in G. The multipoint evaluation method in the exponential form using SS multiplication takes $O(d \log^2 d \log\log d)$ operations in G. □

4. *Generalized birthday problem: non-uniform distribution*

Consider a function $f(x)$ on $\mathbb{F}_p$, with image size N. If one evaluates $f(x)$ at random points repeatedly, one eventually has a collision $f(x_i) = f(x_j)$ for $i \neq j$, since its image is finite. Assuming that f behaves like a random function, the birthday paradox implies a high probability that the collision occurs in $O(\sqrt{N})$ steps.

For the function to behave like a random function it would be necessary that the preimages are all of a similar size. This is not always the case if the function is given by a random polynomial of degree d. For the efficiency of our algorithm, we hope to find a collision faster than $O(\sqrt{N})$. This leads us to consider a birthday problem that applies when the sampling probability is not uniformly distributed.

Suppose that we have N bins, numbered from one to N. For $k = 1, 2, \ldots, N$, let w_k be the probability that a randomly sampled ball is put into the bin k. We are interested in finding the probability of a bin containing at least two different balls.

This kind of problem has been discussed in a number of contexts, for example, [9, 18, 21]. The expected number of trials until a collision has been precisely determined, and is given by $\sqrt{\pi/(2\sum_k w_k^2)} + O(N^{1/4})$. However, such analyses only apply when the probability w_k is bounded by c/N, where c is a constant that is independent of N. In our case, the probability w_k can take values up to d/N, where $d = N^{1/3}$. Therefore, we present an analysis of a non-uniform birthday problem, in which the probabilities are arbitrary. Our analysis shows that a collision occurs with a non-negligible probability in $O(\sqrt{1/\sum_k w_k^2})$ samplings, for any probability distribution of w_k.

Let S_r be the probability that a collision occurs in r trials. Define $E_k^{(r)}$ as the event that a collision occurs in the bin k after r trials. Then, by the Bonferroni inequality[†],

$$\begin{aligned} S_r = \Pr(E_1^{(r)} \cup \ldots \cup E_N^{(r)}) &= \sum_{i=1}^{N} (-1)^{i+1} \sum_{1 \leqslant k_1 < k_2 < \ldots < k_i \leqslant N} \Pr(E_{k_1}^{(r)} \cap \ldots \cap E_{k_i}^{(r)}) \\ &\geqslant \sum_{k=1}^{N} \Pr(E_k^{(r)}) - \sum_{1 \leqslant k < \ell \leqslant N} \Pr(E_k^{(r)} \cap E_\ell^{(r)}). \end{aligned}$$

Unless there is no ambiguity, we will omit the superscript (r) in $E_k^{(r)}$. We first determine a lower bound for $\Pr(E_k)$.

Definition 1. For $k \in \{1, 2, \ldots, N\}$, $B_{r,k}^{(i)}$ is the set of vectors $\vec{b} = (b_1, \ldots, b_r) \in \{1, 2, \ldots, N\}^r$ such that the number of indexes of j satisfying $b_j = k$ is equal to i.

Lemma 4.1. *For a positive integer N, and some $k \in \{1, 2, \ldots, N\}$, let w_k be the probability that a randomly sampled ball is put into the bin k. Let E_k be the event that the bin k contains at least two different balls after $r \geqslant 2$ samplings. Then, the probability of E_k is bounded below by*

$$\Pr(E_k) \geqslant \frac{(r-1)r}{2} \cdot w_k^2 \left\{ 1 - (r-1)\left(1 - \frac{2}{r}\right) w_k \right\}.$$

Proof. With the notation from Definition 1,

$$\Pr(E_k) = \sum_{i \geqslant 2} \sum_{\vec{b} \in B_{r,k}^{(i)}} w_{b_1} \ldots w_{b_r} = 1 - \left(\sum_{\vec{b} \in B_{r,k}^{(1)}} w_{b_1} \ldots w_{b_r} + \sum_{\vec{b} \in B_{r,k}^{(0)}} w_{b_1} \ldots w_{b_r} \right).$$

The summation $\sum_{\vec{b} \in B_{r,k}^{(1)}} w_{b_1} \ldots w_{b_r}$ gives the probability that only one ball is put into the bin k within r trials, so $\sum_{\vec{b} \in B_{r,k}^{(1)}} w_{b_1} \ldots w_{b_r} = r \cdot w_k \cdot (1 - w_k)^{r-1}$. Similarly, $\sum_{\vec{b} \in B_{r,k}^{(0)}} w_{b_1} \ldots w_{b_r} = (1 - w_k)^r$, because this is the probability that no ball is thrown into the bin k within r trials. It follows that

$$\Pr(E_k) = 1 - (r \cdot w_k \cdot (1 - w_k)^{r-1} + (1 - w_k)^r) = 1 - (1 - w_k)^{r-1} \cdot (1 + (r-1) w_k).$$

On the other hand, for $r \geqslant 2$,

$$\begin{aligned} 1 - (1 - w_k)^{r-1} \cdot (1 + (r-1) w_k) &\geqslant 1 - \left(1 - (r-1) w_k + \binom{r-1}{2} w_k^2 \right) \cdot (1 + (r-1) w_k) \\ &\geqslant \frac{(r-1)r}{2} \cdot w_k^2 - \frac{(r-1)^2 (r-2)}{2} \cdot w_k^3 \\ &= \frac{(r-1)r}{2} \cdot w_k^2 \left\{ 1 - (r-1)\left(1 - \frac{2}{r}\right) w_k \right\}. \end{aligned}$$

[†] It is easy to check the lower bound inequality. Assume that $\Pr[E_1 \cup E_2] \geqslant \Pr[E_1] + \Pr[E_2] - \Pr[E_1 \cap E_2]$ (indeed the equality holds in this case). Then to see that

$$\begin{aligned} \Pr[(E_1 \cup E_2) \cup E_3] &= \Pr[E_1 \cup E_2] + \Pr[E_3] - \Pr[(E_1 \cup E_2) \cap E_3] \\ &\geqslant \Pr[E_1] + \Pr[E_2] + \Pr[E_3] - \Pr[E_1 \cap E_2] - \Pr[E_1 \cap E_3] - \Pr[E_2 \cap E_3], \end{aligned}$$

it is enough to check that

$$\Pr[(E_1 \cup E_2) \cap E_3] = \Pr[(E_1 \cap E_3) \cup (E_2 \cap E_3)] \leqslant \Pr[E_1 \cap E_3] + \Pr[E_2 \cap E_3].$$

Now apply the induction on N.

In the first inequality, we used the fact that $(1-x)^n \leqslant 1-nx+\binom{n}{2}x^2$ for $0\leqslant x\leqslant 1$ and $n\geqslant 1$. □

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

LEMMA 4.2. *We employ the same notation as in Lemma 4.1. Assume that $k\neq\ell$. Then,*

$$\Pr(E_k\cap E_\ell)=\sum_{i,j\geqslant 2}\sum_{\vec{b}\in B^{(i)}_{r,k}\cap B^{(j)}_{r,\ell}} w_{b_1}\dots w_{b_r}\leqslant \binom{r}{2}\cdot\binom{r-2}{2}\cdot w_k^2\cdot w_\ell^2.$$

Proof. For $\vec{b}=(b_1,\dots,b_r)\in B^{(i)}_{r,k}\cap B^{(j)}_{r,\ell}$, with $i\geqslant 2, j\geqslant 2$, there exist $i_1\neq i_2$ and $j_1\neq j_2$ such that $b_{i_1}=b_{i_2}=k$ and $b_{j_1}=b_{j_2}=\ell$. In this case, we have $w_{b_1}\dots w_{b_r}\leqslant w_k^2\cdot w_\ell^2$. The value $\binom{r}{2}$ indicates the possible number of two positions for k, and $\binom{r-2}{2}$ denotes the possible number of the other two positions for ℓ. □

Using the above results, we can prove Theorem 2.2.

Proof of Theorem 2.2. Assume that $r\leqslant 1/2W$. Then $(r-1)(1-2/r)w_k\leqslant(r-1)w_k\leqslant(r-1)/2r\leqslant\frac{1}{2}$, for all k. This yields that $\Pr(E_k)\geqslant(r-1)r/4\cdot w_k^2$, using Lemma 4.1. Let $B(r):=((r-1)r/2)\sum_{k=1}^{N}w_k^2$. Then

$$\begin{aligned}S_r &\geqslant \sum_{1\leqslant k\leqslant N}\Pr(E_k)-\sum_{1\leqslant k<\ell\leqslant N}\Pr(E_k\cap E_\ell)\geqslant\frac{B(r)}{2}-\frac{r^2(r-1)^2}{4}\cdot\sum_{1\leqslant k<\ell\leqslant N}w_k^2w_\ell^2\\ &=\frac{B(r)}{2}-\frac{r^2(r-1)^2}{8}\cdot\left\{\left(\sum_{1\leqslant k\leqslant N}w_k^2\right)^2-\left(\sum_{1\leqslant k\leqslant N}w_k^4\right)\right\}\geqslant\frac{B(r)}{2}-\frac{B(r)^2}{2}.\end{aligned}\tag{4.1}$$

The last term, $(B(r)/2)(1-B(r))$, is maximized by $\frac{1}{8}$ when $r=r_0$, such that $B(r_0)=\frac{1}{2}$. That is, $r_0(r_0-1)=D$ or, equivalently, $r_0=\sqrt{D+\frac{1}{4}}+\frac{1}{2}$.

If $\lceil r_0\rceil\leqslant 1/2W$, then the above inequality holds, so $S_{\lceil r_0\rceil}\geqslant[\frac{1}{2}B(1-B)]_{r=\lceil r_0\rceil}\geqslant[\frac{1}{2}B(1-B)]_{r=r_0+1}$. The last inequality comes from the fact that $\frac{1}{2}B(1-B)$ is decreasing for $1/2\leqslant B$ or, equivalently, for $r\geqslant r_0$. If $D\geqslant 20$ (equivalently, $r_0\geqslant 5$), then

$$[B(r)\cdot D]_{r=r_0+1}=\frac{(r_0+1)r_0}{2}=\frac{1}{2}\left(D+1+2\sqrt{D+\frac{1}{4}}\right)\leqslant\frac{3}{4}\cdot D.$$

Thus, $S_{\lceil r_0\rceil}\geqslant[\frac{1}{2}B(1-B)]_{r=\lceil r_0\rceil}\geqslant[\frac{1}{2}B(1-B)]_{r=r_0+1}\geqslant[\frac{1}{2}B(1-B)]_{B=3/4}=\frac{3}{32}$. Since S_r increases as r grows, $S_r\geqslant\frac{3}{32}$ for $r\geqslant r_0$.

On the other hand, assume that $1/2W\leqslant\lceil r_0\rceil$ and r_0 is not an integer. Then $1/2W-1\leqslant\lfloor r_0\rfloor\leqslant r_0$. Furthermore,

$$\begin{aligned}\frac{1}{2}&\geqslant[B(r)]_{r=(1/2W)-1}=\left(\frac{1}{8W^2}-\frac{3}{4W}+1\right)\cdot\sum_k w_k^2\geqslant\left(\frac{1}{8W^2}-\frac{3}{4W}+1\right)\cdot W^2\\ &=\frac{1}{8}-\frac{3W}{4}+W^2\geqslant\frac{1}{8}-\frac{3}{4}\cdot\frac{1}{8}+\left(\frac{1}{8}\right)^2\geqslant\frac{1}{22},\end{aligned}$$

where the first inequality comes from the fact that $B(r)\leqslant\frac{1}{2}$ if and only if $r(r-1)\leqslant 1/(\sum_k w_k^2)$ if and only if $\frac{1}{2}-\sqrt{D+\frac{1}{4}}\leqslant r\leqslant\frac{1}{2}+\sqrt{D+\frac{1}{4}}$. Because $\frac{1}{2}B(1-B)$ is increasing for $B\leqslant\frac{1}{2}$,

$S_{\lfloor r_0\rfloor} \geqslant S_{\lceil (1/2W)-1\rceil} \geqslant [\frac{1}{2}B(1-B)]_{r=\lceil (1/2W)-1\rceil} \geqslant [\frac{1}{2}B(1-B)]_{r=(1/2W)-1} \geqslant \frac{1}{2}\cdot\frac{1}{22}(1-\frac{1}{2}) \geqslant \frac{1}{88}$. Therefore, $S_{\lceil r_0\rceil} \geqslant S_{\lfloor r_0\rfloor} \geqslant \frac{1}{88}$.

If $1/2W \leqslant \lceil r_0\rceil$ and r_0 is an integer, then $1/2W \leqslant \lceil r_0\rceil = r_0$. Then, similarly to the above, we have $S_{\lceil r_0\rceil} = S_{r_0} \geqslant S_{\lfloor 1/2W\rfloor} \geqslant [\frac{1}{2}B(1-B)]_{r=\lfloor 1/2W\rfloor} \geqslant [\frac{1}{2}B(1-B)]_{r=(1/2W)-1} \geqslant \frac{1}{88}$. □

The above theorem shows that a collision occurs with a non-negligible probability after approximately $1/\sqrt{\sum_k w_k^2}$ trials, although the probabilities are arbitrarily distributed. In the following, we consider several examples.

Example 1. In the case where $w_k = O(1/N)$ for all k, there is a non-negligible probability that a collision occurs after $\Omega(\sqrt{N})$ trials, as in the case of the usual birthday paradox.

Theorem 2.2 asserts that a collision occurs with a high probability after $O(\sqrt{1/(\sum_k w_k^2})$) trials. Specifically, it was shown in [9] that when $w_k = O(1/N)$, the expected number of trials until a collision occurs is given by $\sqrt{\pi/(2\sum_k w_k^2)} + O(N^{1/4})$, as $N \to \infty$.

Example 2. The proposed theorem applies, even when $w_k = \Omega(1/N)$, for some k. In the proof of Theorem 2.2, we have shown that $S_{1/2W} \geqslant \frac{1}{64}$ for $W = \max_k\{w_k\}$. This is meaningful when $W = \Omega(1/\sqrt{N})$, because a collision is guaranteed in $O(1/W)$ trials, which is faster than the usual expectation of the birthday paradox.

Example 3. Consider the birthday problem given by a polynomial $f \in \mathbb{F}_p[x]$, as in Theorem 2.3. Suppose that the probabilities are given by

$$(w_1,\ldots,w_v) = \Bigg(\underbrace{\frac{1}{p},\ldots,\frac{1}{p}}_{R_1},\underbrace{\frac{2}{p},\ldots,\frac{2}{p}}_{R_2},\ldots,\underbrace{\frac{d}{p},\ldots,\frac{d}{p}}_{R_d}\Bigg).$$

The size of the value set of f is $\sum_i R_i$, and a rough estimate of the birthday paradox suggests that a collision occurs in $O(\sqrt{\sum_i R_i})$. However, a collision can be found in $O(N/\sqrt{\sum_i i^2 R_i}) \leqslant O(\sqrt{\sum_i R_i})$, by Theorem 2.2. The inequality comes from the Cauchy–Schwartz inequality.

5. *Polynomials for the proposed algorithm*

In the rest of this paper, we assume that d is relatively prime to p.

5.1. *Substitution polynomials*

Let $f(x,y) \in \mathbb{F}[x,y]$ be an irreducible bivariate polynomial, defined over a field $\mathbb{F}$. The polynomial f is said to be *absolutely irreducible* if it is also irreducible over the algebraic closure. For a polynomial $f(x)$, one defines the *substitution polynomial of* f as the bivariate polynomial $f(x) - f(y)$.

For the algorithm to be efficient, one requires a polynomial f with as large a value of ρ_f as possible. In the following lemma, we observe that ρ_f is closely related to the number of absolutely irreducible factors of the substitution polynomial $f(x) - f(y)$.

Lemma 5.1 (Weil's bound [24]). *Let $f \in \mathbb{F}_p[x]$ be a polynomial of degree d, and let τ_f be the number of absolutely irreducible factors of the substitution polynomial of f. Then, $\tau_f p - d^2\sqrt{p} \leqslant \rho_f \leqslant \tau_f p + d^2\sqrt{p}$.*

From Weil's bound, for $d < p^{1/4}$, we see that the complexity of the proposed algorithm becomes

$$\widetilde{O}\Big(\sqrt{p^2/\rho_f} + d\Big) \sim \widetilde{O}\Big(\sqrt{p/\tau_f} + d\Big).$$

Therefore, Lemma 5.1 reduces the proposed algorithm to finding a polynomial whose substitution polynomial has as many absolutely irreducible factors as possible. In the following subsections, we will discuss an upper bound for the number τ_f, and then try to find polynomials that attain this bound.

5.2. *An upper bound of the number of absolutely irreducible factors*

We note that there exist polynomials for which the substitution polynomial factorizes into all linear absolutely irreducible factors (that is, $\tau_f = d$) when $d \mid \Phi_1(p) = (p-1)$, or all quadratics with one or two linear exceptions (that is, $\tau_f \approx d/2$) when $d \mid \Phi_2(p) = (p+1)$ [**10**, **15**]. From this observation, we attempt to find a polynomial in the case where $d \mid \Phi_k(p)$, whose substitution polynomial factorizes into all k-degree factors except for a few small degree factors. We show that the substitution polynomial of any polynomial cannot yield absolutely irreducible cubic factors in the case where $k = 3$, by using the same idea as in some previous papers (see, for example, [**10**–**13**]). This shows that we cannot achieve $\tau_f \approx d/3$ in the case where $d \mid \Phi_3(p) = (p^2+p+1)$.

Assume that the factorization of $f(x) - f(y)$ into irreducible factors over $\mathbb{F}_p$ is given by

$$f(x) - f(y) = g_1(x,y) \ldots g_s(x,y).$$

Let $g_i(x,y) = h_{i,d_i} + h_{i,d_i-1} + \ldots + h_{i,1} + h_{i,0}$, where $h_{i,j} \in \mathbb{F}_p[x,y]$ is the homogeneous part of degree j in $g_i(x,y)$, and d_i denotes the highest degree of $g_i(x,y)$. Furthermore, we assume that $g_s(x,y) = x - y$.

As an aside, we also give alternative proofs for Lemma 5.2 and Theorem 5.3 in the appendix.

LEMMA 5.2. *Let d be a positive integer, dividing $\Phi_k(p)$ for prime k, and let ζ be a primitive dth root of unity in $\mathbb{F}_{p^k}$. Then, the following holds. Either $\zeta^i \in \mathbb{F}_{p^k} \backslash \mathbb{F}_p$ for all $i \not\equiv 0 \pmod{d}$, if $d \equiv 1 \pmod{k}$, or only $\zeta^{(i/k)\cdot d}$ for $i = 0, 1, \ldots, k-1$ are in $\mathbb{F}_p$, if $d \equiv 0 \pmod{k}$. Note that there exists no positive integer d dividing $\Phi_k(p)$ if $d \not\equiv 0, 1 \pmod{k}$.*

Proof. Note that $\zeta^i \in \mathbb{F}_p$ if and only if $\zeta^{i(p-1)} = 1$ if and only if $i(p-1) \equiv 0 \pmod{d}$. The number of such i is equal to $\gcd(d, p-1)$. The value of $\gcd(d, p-1)$ divides

$$\gcd(\Phi_k(p), p-1) = \gcd(p^{k-1} + \ldots + p + 1, p-1) = \gcd(p-1, k),$$

which can only be 1 or k for prime k.

If $\gcd(d, p-1) = k$, then $d \equiv 0 \pmod{k}$ and all the kth roots of unity, $\zeta^{(i/k)\cdot d}$, lie in $\mathbb{F}_p$. If $\gcd(d, p-1) = 1$, then only $\zeta^0 = 1$ lies in $\mathbb{F}_p$, and all $\zeta^i \in \mathbb{F}_{p^k} \backslash \mathbb{F}_p$ for $i \neq 0$ must form conjugate k-tuples

$$(\zeta^i, (\zeta^i)^p, (\zeta^i)^{p^2}, \ldots, (\zeta^i)^{p^{k-1}}),$$

which is only possible when $d - 1 \equiv 0 \pmod{k}$. Otherwise, if $d \not\equiv 0, 1 \pmod{k}$, d cannot divide $\Phi_k(p)$. □

In the following theorem, we give an upper bound for τ_f.

THEOREM 5.3. *Let $f \in \mathbb{F}_p[x]$ be a polynomial of degree d. Let k be a prime number. Assume that $d \mid \Phi_k(p)$. Let τ_f be the number of absolutely irreducible factors in the factorization of $f(x) - f(y)$. Then, $\tau_f \leqslant \sum_{D \mid d} (\varphi(D)/\mathrm{ord}_D(p))$. In particular, when k is the prime, either*

$$\tau_f \leqslant \frac{d-1}{k} + 1 \quad \text{for } d \equiv 1 \pmod{k} \quad \text{or} \quad \tau_f \leqslant \frac{d-k}{k} + k \quad \text{for } d \equiv 0 \pmod{k}.$$

Proof. Consider

$$f(x)-f(y)=(x^d-y^d)+a_{d-2}(x^{d-2}-y^{d-2})+\ldots+a_2(x^2-y^2)+a_1(x-y)=g_1\ldots g_s.$$

Comparing the highest homogeneous term gives

$$x^d-y^d=h_{1,d_1}\ldots h_{s,d_s},\quad \text{where } h_{i,d_i}\in\mathbb{F}_p[x,y].$$

Since $x^d-y^d=\prod_{D|d}\Phi_D(x,y)$ and $\Phi_D(x,y)$ factorizes into $\varphi(D)/\mathrm{ord}_D(p)$ distinct irreducible factors of degree $\mathrm{ord}_D(p)$, we have at most $\sum_{D|d}(\varphi(D)/\mathrm{ord}_D(p))$ absolutely irreducible factors. Here, $\mathrm{ord}_D(p)$ denotes the multiplicative order of p modulo D.

Let ζ be a primitive dth root of unity in $\mathbb{F}_{p^k}$. For the prime k, x^d-y^d has irreducible factors (over $\mathbb{F}_p$) of either a linear factor, $(x-\zeta^i y)$, for $\zeta^i\in\mathbb{F}_p$, or a degree-k factor,

$$(x-\zeta^i y)(x-\zeta^{i\cdot p}y)\ldots(x-\zeta^{i\cdot p^{k-1}}y),$$

for $\zeta^i\in\mathbb{F}_{p^k}\backslash\mathbb{F}_p$. Therefore, by Lemma 5.2, the number of irreducible factors of x^d-y^d is either $(d-1)/k+1$ for $d\equiv 1\,(\mathrm{mod}\,k)$ or $(d-k)/k+k$ for $d\equiv 0\,(\mathrm{mod}\,k)$. Because the factor g_i is determined by its highest degree term h_{i,d_i}, the number of absolutely irreducible factors is less than the number of irreducible factors of x^d-y^d. □

5.3. *Several examples*

In this section, we present several polynomials of degree $d<p$ that achieve the upper bound of τ_f, in the case where $d\mid\Phi_1(p)$ and $d\mid\Phi_2(p)$. In the case where $d\mid\Phi_3(p)$, no polynomial can attain the upper bound.

5.3.1. *Case* 1: $d\mid\Phi_1(p)=p-1$. In this case, the possible number of irreducible factors is at most d. Consider $f(x)=x^d$, where $d\mid(p-1)$. Then, a primitive dth root of unity ζ exists in $\mathbb{F}_p$, and $f(x)-f(y)$ has d absolutely irreducible linear factors over $\mathbb{F}_p$, because the factorization is given by

$$f(x)-f(y)=\prod_{i=1}^{d}(x-\zeta^i y).$$

For a fixed non-zero y, $f(x)=f(y)$ if and only if $x=\zeta^i y$ for each $i=1,\ldots,d$, so the map $x\mapsto f(x)$ is a d-to-one function, except on $x=0$. Finally, $\rho_f=R_1+d^2\cdot R_d=1+d^2\cdot(p-1)/d=1+d(p-1)$.

REMARK 2. By applying Theorem 2.3, with the polynomial $f(x)=x^d$ such that $d\mid(p-1)$, we can solve for the discrete log α in $\widetilde{O}(\sqrt{p/d}+d)$ group operations, which can be lowered by $\widetilde{O}(p^{1/3})$ when $d=p^{1/3}$. Note that a polynomial of form $f(x)=a(x+b)^d+c$ suggests the same asymptotic complexity, because cardinality of the value set is not altered by translations.

5.3.2. *Case* 2: $d\mid\Phi_2(p)=p+1$. In this case, the number of possible absolutely irreducible factors is at most $\lfloor(d+2)/2\rfloor$. Consider the Dickson polynomial of degree d. For a non-zero $a\in\mathbb{F}_p$, the Dickson polynomial is defined as

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

The following lemma shows that the substitution polynomial of the Dickson polynomial has exactly $\lfloor(d+2)/2\rfloor$ absolutely irreducible factors, and presents the exact value of ρ_f.

LEMMA 5.4 [10, 15]. *Assume that $d \mid (p+1)$, and that ζ is a primitive dth root of unity in $\mathbb{F}_{p^2}$. It then holds that*

$$D_d(x,a) - D_d(y,a) = (x^t - y^t) \prod_{i=1}^{\lfloor (d-1)/2 \rfloor} (x^2 - (\zeta^i + \zeta^{-i})xy + y^2 + a(\zeta^{2i} + \zeta^{-2i} - 2)),$$

where $t = 1$ for odd d and $t = 2$ for even d. The value of $\rho_f =$ is given by

$$\rho_f = \frac{(d+1)p}{2} + O(d^2).$$

REMARK 3. Applying Theorem 2.3 with the Dickson polynomial $D_d(x,a)$, where $d \mid (p+1)$, the discrete log α can be recovered within $\widetilde{O}(\sqrt{p/2d} + d)$ group operations for $d < p^{1/2}$. It can be lowered to $\widetilde{O}(p^{1/3})$ when $d = p^{1/3}$.

5.3.3. *Case* 3: $d \mid \Phi_3(p) = p^2 + p + 1$. In this case, τ_f is bounded above by $(d-1)/3 + 1$ for $d \equiv 1 \pmod 3$, and $(d-3)/3 + 3$ for $d \equiv 0 \pmod 3$. This type of polynomial only appears when the factorization of $f(x) - f(y)$ is given by

$$f(x) - f(y) = (x^t - y^t) \prod_{i=1}^{s-1} g_i(x,y),$$

where each g_i is an absolutely irreducible cubic factor ($t = 1$ or $t = 3$, depending on the residue class of d modulo three). However, the following theorem asserts that such a polynomial does not exist, which is a direct consequence of [12, Lemma 6].

THEOREM 5.5. *Let ζ be a primitive dth root of unity in $\mathbb{F}_{p^3}$, and assume that $f(x) \in \mathbb{F}_p[x]$ is a polynomial of degree d. Then, $f(x) - f(y)$ cannot have an absolutely irreducible cubic factor.*

Proof. Write $f(x) = x^d + a_{d-1}x^{d-1} + \ldots + a_1 x + a_0 \in \mathbb{F}_p[x]$. Without loss of generality, we assume that $a_{d-1} = 0$. Similarly, as before, denote

$$\frac{f(x) - f(y)}{x - y} = \frac{x^d - y^d}{x - y} + a_{d-2}\frac{x^{d-2} - y^{d-2}}{x - y} + \ldots + a_2 \frac{x^2 - y^2}{x - y} + a_1 = g_1(x,y) \ldots g_{s-1}(x,y).$$

If $g_i(x,y)$ is an irreducible cubic factor, then the highest homogenous term should be of form $(x - \xi y)(x - \xi^p y)(x - \xi^{p^2} y)$, where $\xi := \zeta^j$ for some j. On the other hand, by [12, Lemma 6], the coefficient of y^3 in the cubic factors must be equal to 1, which contradicts the fact that the coefficient is -1 in $(x - \xi y)(x - \xi^p y)(x - \xi^{p^2} y)$. □

6. *Conclusion*

We have proposed a new algorithm for solving the DLPwAI. The algorithm has a running time of $\widetilde{O}(p/\sqrt{\rho_f} + d) = \widetilde{O}(\sqrt{p/\tau_f} + d)$ group exponentiations, for a chosen polynomial $f \in \mathbb{F}_p[x]$ of degree d. Therefore, it reduces the DLPwAI to a problem of finding polynomials with a large value of ρ_f or τ_f.

It remains an open problem to find a polynomial with sufficiently large τ_f so that the proposed algorithm has a complexity of $O(\sqrt{p/d})$ as the lower bound in the generic group model. For example, we can find such polynomials in the case where $d \mid (p \pm 1)$.

Regarding the birthday problem, it would be interesting to determine the expected number of trials until a collision for arbitrary probability distributions.

Appendix. *Another proof of Lemma 5.2 and Theorem 5.3*

We begin by stating some notation. Assume that a primitive dth root of unity ζ lies in $\mathbb{F}_{p^k}$, where k is the smallest integer satisfying the condition. For $\tilde{k} \mid k$, we define the following.

- $D(\tilde{k})$ is the number of $i \in \{1, 2, \ldots, d\}$ satisfying the condition that ζ^i lies in $\mathbb{F}_{p^{\tilde{k}}}$.
- $N(\tilde{k})$ is the number of $i \in \{1, 2, \ldots, d\}$ satisfying the condition that ζ^i exactly lies in $\mathbb{F}_{p^{\tilde{k}}}$, and not in any proper subfield.

Proof of Lemma 5.2 and Theorem 5.3. It suffices to find the number of irreducible factors of $x^d - 1$ over $\mathbb{F}_p$. If ζ^i is in $\mathbb{F}_{p^{\tilde{k}}}$ and not in any proper subfield, then the minimal polynomial of ζ^i is of degree $\tilde{k}$. Therefore, $x^d - 1$ factorizes into $\sum_{\tilde{k}|k} (N(\tilde{k})/\tilde{k})$ irreducible factors.

Now, we can easily check that $D(\tilde{k}) = \gcd(d, p^{\tilde{k}} - 1)$, since $\zeta^i \in \mathbb{F}_{p^{\tilde{k}}}$ if and only if $\zeta^{i(p^{\tilde{k}}-1)} = 1$ if and only if $i(p^{\tilde{k}} - 1) \equiv 0 \pmod d$. From the definitions, $D(\tilde{k}) = \sum_{\ell|\tilde{k}} N(\ell)$, so the Möbius inversion formula suggests that

$$N(\tilde{k}) = \sum_{\ell|\tilde{k}} \mu\left(\frac{\tilde{k}}{\ell}\right) \cdot D(\tilde{k}) = \sum_{\ell|\tilde{k}} \mu\left(\frac{\tilde{k}}{\ell}\right) \cdot \gcd(d, p^{\tilde{k}} - 1),$$

where $\mu(n)$ is the Möbius function.

For prime k,

$$\begin{aligned}\sum_{\tilde{k}|k} \frac{N(\tilde{k})}{\tilde{k}} &= N(1) + \frac{N(k)}{k} = \gcd(d, p-1) + \frac{\gcd(d, p^k - 1) - \gcd(d, p-1)}{k} \\ &= \gcd(d, p-1) + \frac{d - \gcd(d, p-1)}{k}.\end{aligned}$$

Since $N(k) = d - \gcd(d, p-1)$ must be a multiple of k, and $\gcd(d, p-1)$ can only be either one or k, d modulo k can only be either one or zero. □

Acknowledgements. The authors wish to express their sincere gratitude to S. Galbraith for handling our paper. They also would like to thank H. Ryu, J. H. Seo, Y. S. Song, M. Tibouchi and M. Zieve for their useful discussions. They also appreciate the help of the anonymous reviewers who further improved the paper.

References

1. D. Boneh and X. Boyen, ‘Efficient selective-ID secure identity-based encryption without random oracles’, *Advances in cryptology - EUROCRYPT 2004*, Lecture Notes in Computer Science 3027 (eds C. Cachin and J. Camenisch; Springer, Berlin, 2004) 223–238.
2. D. Boneh and X. Boyen, ‘Short signatures without random oracles’, *Advances in cryptology - EUROCRYPT 2004*, Lecture Notes in Computer Science 3027 (eds C. Cachin and J. Camenisch; Springer, Berlin, 2004) 56–73.
3. D. Boneh, C. Gentry and B. Waters, ‘Collusion resistant broadcast encryption with short ciphertexts and private keys’, *Advances in cryptology - CRYPTO 2005*, Lecture Notes in Computer Science 3621 (ed. V. Shoup; Springer, Berlin, 2005) 258–275.
4. D. R. L. Brown and R. P. Gallant, ‘The static Diffie–Hellman problem’, IACR Cryptology ePrint Archive (2004), http://eprint.iacr.org/2004/306.
5. J. H. Cheon, ‘Security analysis of the strong Diffie–Hellman problem’, *Advances in cryptology - EUROCRYPT 2006*, Lecture Notes in Computer Science 4004 (ed. S. Vaudenay; Springer, Berlin, 2006) 1–11.
6. J. H. Cheon, ‘Discrete logarithm problems with auxiliary inputs’, *J. Cryptology* 23 (2010) 457–476.
7. J. H. Cheon and T. Kim, ‘Discrete logarithm with auxiliary inputs’, *MSJ-KMS Joint Meeting 2012* (2012).

8. J. H. Cheon, T. Kim and Y. S. Song, 'A group action on $\mathbb{F}_p^{\times}$ and the generalized DLP with auxiliary inputs', *Selected areas in cryptography 2013*, Lecture Notes in Computer Science 8282 (eds T. Lange, K. E. Lauter and P. Lisonek; Springer, Berlin, 2013) 121–135.

9. S. D. Galbraith and M. Holmes, 'A non-uniform birthday problem with applications to discrete logarithms', *Discrete Appl. Math.* 160 (2012) 1547–1560.

10. J. Gomez-Calderon and D. J. Madden, 'Polynomials with small value set over finite fields', *J. Number Theory* 28 (1988) 167–188.

11. J. Gomez-Calderon, 'On the cardinality of value set of polynomials with coefficients in a finite field', *Proc. Japan Acad. Ser. A Math. Sci.* 68 (1992) 338–340.

12. J. Gomez-Calderon, 'The third-order factorable core of polynomials over finite fields', *Proc. Japan Acad. Ser. A Math. Sci.* 74 (1998) 16–19.

13. D. R. Hayes, 'A geometric approach to permutation polynomials over a finite field', *Duke Math. J.* 34 (1967) 293–305.

14. M. Kim, J. H. Cheon and I.-S. Lee, 'Analysis on a generalized algorithm for the strong discrete logarithm problem with auxiliary inputs', *Math. Comp.* 83 (2014) 1993–2004.

15. D. A. Mit'kin, 'Polynomials with minimal set of values and the equation $f(x) = f(y)$ in a finite prime field', *Mat. Zametki* 38 (1985) 3–14.

16. S. Mitsunari, R. Sakai and M. Kasahara, 'A new traitor tracing', *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E85-A (2002) 481–484.

17. P. Mohassel, 'Fast computation on encrypted polynomials and applications', *CANS*, Lecture Notes in Computer Science 7092 (eds D. Lin, G. Tsudik and X. Wang; Springer, 2011) 234–254.

18. K. Nishimura and M. Sibuya, 'Occupancy with two types of balls', *Ann. Inst. Statist. Math.* 40 (1988) 77–91.

19. J. M. Pollard, 'Monte Carlo methods for index computation $(\bmod p)$', *Math. Comp.* 32 (1978) 918–924.

20. T. Satoh, On generalization of Cheon's algorithm, IACR Cryptology ePrint Archive (2009), http://eprint.iacr.org/2009/058.

21. B. I. Selivanov, 'On waiting time in the scheme of random allocation of coloured particies', *Discrete Math. Appl.* 5 (1955) 73–82.

22. S. Uchiyama, 'Note on the mean value of $v(f)$', *Proc. Japan Acad.* 31 (1955) 199–201.

23. J. von zur Gathen and J. Gerhard, *Modern computer algebra* (Cambridge University Press, Cambridge, 2003).

24. A. Weil, *Sur les Courbes algébriques et les variétés qui s'en déduisent*, Actualités Scientifiques et Industrielles 1041 (Hermann & Cie, 1948).

Jung Hee Cheon
Department of Mathematical Sciences
Seoul National University
GwanAkRo 1
Gwanak-Gu
Seoul 151-747
Korea

jhcheon@snu.ac.kr

Taechan Kim
NTT Secure Platform Laboratories
3-9-11
Midori-cho
Musashino-Shi
Tokyo 180-8585
Japan

taechan.kim@lab.ntt.co.jp