MATHEMATICS OF COMPUTATION
Volume 83, Number 288, July 2014, Pages 1993–2004
S 0025-5718(2014)02813-5
Article electronically published on February 11, 2014

ANALYSIS ON A GENERALIZED ALGORITHM FOR THE STRONG DISCRETE LOGARITHM PROBLEM WITH AUXILIARY INPUTS

MINKYU KIM, JUNG HEE CHEON, AND IN-SOK LEE

Abstract. We investigate a recently proposed algorithm solving the strong discrete logarithm problem with auxiliary inputs, and show that this algorithm in general is not more efficient than ordinary discrete-logarithm-solving algorithms such as Pollard's rho method, by analyzing a lower bound on the sum of digits of integers.

1. Introduction

The strong discrete logarithm problem with ℓ auxiliary inputs (ℓ-SDL) asks for the discrete logarithm (DL) $\alpha \in \mathbb{F}_p$, given g, g^α as well as auxiliary inputs g^{α^i} for $i = 2, \ldots, \ell$, where g is an element of prime order p in an abelian group $\mathbb{G}$. This problem together with similar problems for the Diffie-Hellman (DH) problem [11] as the ℓ-Strong DH, the Bilinear DH Inversion, and the Bilinear DH Exponent problem, were originally believed to be as difficult as the ordinary DL or DH problem. Based on the intractability of these problems, a variety of cryptosystems have been designed such as traitor tracing schemes [18], ID-based encryptions [1,3], short group signatures [4], broadcast encryptions [6], short signatures [2], and blind signatures [20], to name a few.

However, it turned out that they might have reduced complexity when $p-1$ or $p+1$ have an appropriate divisor [5, 8, 9]. More precisely, it was shown that the d-SDL problem can be solved in $\mathcal{O}\big((p/d)^{1/2}+d^{1/2}\big)$ exponentiations in $\mathbb{G}$ when d is a divisor of $p-1$ and $g, g^\alpha, g^{\alpha^d}$ are given, and if d is a divisor of $p+1$, α can be computed in $\mathcal{O}\big((p/d)^{1/2}+d\big)$ exponentiations in $\mathbb{G}$ using g, g^{α^i} for $i = 1, \ldots, 2d$.

Basically, both of the two algorithms work by implicitly transforming the discrete logarithm α in $\mathbb{F}_p$ into an element $\psi(\alpha)$ in an auxiliary group $\mathbb{H}$, and computing $\psi(\alpha)$ by solving the DL problem over $\mathbb{H}$, where the DL problem can be solved more efficiently than $\mathbb{G}$. The $p-1$ algorithm uses the mapping $\alpha \mapsto \alpha^d$ and the subgroup of order $(p-1)/d$ of $\mathbb{F}_p^\times$ for an auxiliary group. The $p+1$ algorithm employs the mapping $\alpha \mapsto (\alpha+\theta)^{(p-1)\cdot d}$, $\theta \in \mathbb{F}_{p^2}\backslash\mathbb{F}_p$, and the subgroup of order $(p+1)/d$ of $\mathbb{F}_{p^2}^\times$ as an auxiliary group. In fact, these two algorithms can be regarded as quantitative versions of the well-known reduction from the DL problem to the DH problem due to den Boer [10] and Maurer [15], and it is therefore natural to ask whether one

Received by the editor February 14, 2012 and, in revised form, November 1, 2012.
2010 *Mathematics Subject Classification.* Primary 68Q25; Secondary 11Y16.
Key words and phrases. Discrete logarithm problem, (strong) Diffie-Hellman problem, sum of digits.
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0001243).

can generalize the $p \pm 1$ algorithms to other groups defined over $\mathbb{F}_p$, such as other extension fields of $\mathbb{F}_p$ or elliptic or hyperelliptic curves over $\mathbb{F}_p$.

Recently, Takakazu Satoh [22] proposed a possible generation of the $p \pm 1$ algorithms to a divisor D of $\Phi_n(p)$ using the general linear group $\mathrm{GL}_n(\mathbb{F}_p)$ for an auxiliary group, where Φ_n denotes the n-th cyclotomic polynomial. Although Satoh describes his algorithm in the context of general linear groups, he essentially uses the mapping $\alpha \mapsto (\alpha+\zeta)^{(p^n-1)/D}$, $\zeta \in \mathbb{F}_{p^n}^{\times}$. The main obstacle in this direction of generalization is that the degree of that mapping with respect to α is quite high, and so we need g^{α^i} for quite large i, because the degree is equal to the sum of digits of the p-ary expansion of $(p^n-1)/D$. Satoh tried to overcome this problem by multiplying $(p^n-1)/D$ by integers r with $\gcd(r, D) = 1$ to obtain a new integer with small p-ary digits; if possible, one can map the DL α to the subgroup of order D of $\mathbb{F}_{p^n}^{\times}$ only using g^{α^i} for small i. Satoh's algorithm for $n = 1, 2$ corresponds to the $p \pm 1$ algorithms, and the $p+1$ version in particular requires only g, g^{α^i} for $1 \le i \le d = (p+1)/D$. However, for $n \ge 3$, the generalization did not come with a precise complexity analysis, and so it was not clear whether it was indeed faster than the ordinary DL-solving algrithm such as Pollard's rho method [21].

In fact, the problem of finding a good r, having the property that the number $r \cdot (p^n - 1)/D$ has small p-ary digits, can be thought of as the problem of finding a short vector in some well-defined lattices $\mathcal{L}$. Thus, we may be able to find a good r by applying lattice reduction algorithms to the lattice $\mathcal{L}$, such as LLL [14]. In addition, from the Minkowski inequality on lattices [17], one can get an upper bound on the length of the shortest vector and thus an upper bound on the sum of digits of the integers of the form $r \cdot (p^n-1)/D$. Nevertheless, the problem still remains unsolved, because the upper bound obtained from the Minkowski inequality is not sufficient to deduce the complexity less than that of the ordinary DL-solving algorithms and we do not know how small the length of the shortest vector could be.

This paper settles the open problem in the negative, analyzing the sum of digits of integers. We show that, for any integers t and $s = \sum_{i \ge 0} s_i t^i$ with $s_i \in \mathbb{Z}$, if $\sum_{i \ge 0} s_i x^i \not\equiv 0 \bmod \Phi_m(x)$, then

$$\sum_i |s_i| \ge \big|\gcd\big(s, \Phi_m(t)\big)\big|^{1/\varphi(m)},$$

where φ denotes the Euler totient function. As a result, we deduce that the sum of digits of integers of the form $r \cdot (p^n-1)/D$ including the length of the shortest vector in $\mathcal{L}$ are not too small, and thus Satoh's generalization for $n \ge 3$ is never better than the ordinary DL-solving algorithm.

The rest of the paper is organized as follows. Section 2 includes some preliminaries. In Section 3 we first give a simplification of Satoh's algorithm, showing that the mapping $\alpha \mapsto (\alpha+\zeta)^{(p^n-1)/D}$ can be expressed in terms of polynomials over $\mathbb{F}_p$, and then we proceed to give a rigorous complexity analysis. In Section 4 we come to the conclusion that the complexity of Satoh's generalization, including our algorithm, is always greater than $p^{1/2}$ if $(p^n - 1)/D \not\equiv 0 \bmod \Phi_m(p)$ for some divisor $m \ge 3$ of n. The final section summarizes the results of the present study and discusses future research.

2. Preliminaries

In this section we introduce some notation used throughout this paper.

Let p be a prime number and let $\mathbb{F}_p$ denote the finite field $\{0,\dots,p-1\}$ of p elements equipped with addition and multiplication modulo p. For a given integer ω, we consider the signed p-ary expansion of $\omega = \sum_{i\geq 0} \omega_i p^i$ with $|\omega_i| < p/2$, and define $\mathsf{S}_p(\omega) = \max\{\mathsf{S}_p^+(\omega), \mathsf{S}_p^-(\omega)\}$, where

$$\mathsf{S}_p^+(\omega) = \sum_{w_i>0} w_i \quad \text{and} \quad \mathsf{S}_p^-(\omega) = -\sum_{w_i<0} w_i.$$

Let $n \geq 2$ be an integer. The n-th cyclotomic polynomial is denoted by Φ_n and its degree by $\varphi(n)$, where $\varphi(\cdot)$ is the Euler totient function. We fix the representation of the finite field of p^n elements to $\mathbb{F}_{p^n} = \mathbb{F}_p[x]/\langle \mathfrak{p}(x)\rangle$, by fixing an irreducible polynomial $\mathfrak{p}(x) \in \mathbb{F}_p[x]$ of degree n, and view elements of $\mathbb{F}_{p^n}$ as polynomials over $\mathbb{F}_p$ of degree less than n. We put $\mathbf{x}_i = x^{i-1} + \langle \mathfrak{p}(x)\rangle$ for $1 \leq i \leq n$.

We write $\mathsf{M}(\ell)$ to denote the number of operations in a ring R that are required to multiply two polynomials in $R[x]$ degree less than ℓ. Then polynomials in $R[x,y]$ of degree less than n in x and less than ℓ in y can be multiplied using $\mathcal{O}\big(\mathsf{M}(n\ell)\big)$ operations in R [25, Corollary 8.28]. Thus, we can multiply two elements in $\mathbb{F}_{p^n}$ with $\mathsf{M}(n)$ operations in $\mathbb{F}_p$ and multiply two polynomials in $\mathbb{F}_{p^n}[y]$ of degree less than ℓ with $\mathcal{O}\big(\mathsf{M}(n\ell)\big)$ operations in $\mathbb{F}_p$. Note that currently the fastest algorithms for multiplication of polynomials of degree at most ℓ over a ring R take $\mathcal{O}\big(\ell \log(\ell)\log(\log(\ell))\big)$ operations in R [24]. In this paper we will assume that the multiplication time $\mathsf{M}(\cdot)$ satisfies $\mathsf{M}(n\ell) \geq n \cdot \mathsf{M}(\ell)$ for all positive integers n, ℓ. We note that all the known multiplication algorithms satisfy this property.

3. The algorithm solving the SDL problem

In this section we first analyze the mapping ψ from $\mathbb{F}_p$ to some subgroup $\mathbb{H}$ of $\mathbb{F}_{p^n}^\times$, defined by exponentiation by $(p^n-1)/|\mathbb{H}|$, which is the simplest way to construct an element in $\mathbb{H}$. We then proceed to describe the algorithm, solving the SDL problem in the subsequent subsection.

3.1. Mapping on $\mathbb{F}_p$.

Let ζ_τ denote an element of $\mathbb{F}_{p^n}$ which is not contained in any proper subfield of $\mathbb{F}_{p^n}$. Let ω be an integer. We define a function $\psi_\omega : \mathbb{F}_p \to \mathbb{F}_{p^n},\ y \mapsto (y+\zeta_\tau)^\omega$. This function ψ_ω can be presented as follows.

Lemma 3.1. *Let ω be a positive integer less than p^n and let e denote $\mathsf{S}_p(\omega)$. The function ψ_ω can be written as*

$$\psi_\omega(y) = \big(y+\zeta_\tau\big)^\omega = \frac{\hat{f}_1(y)\,\mathbf{x}_1 + \cdots + \hat{f}_n(y)\,\mathbf{x}_n}{\check{f}_1(y)\,\mathbf{x}_1 + \cdots + \check{f}_n(y)\,\mathbf{x}_n}, \tag{3.1}$$

where $\hat{f}_j, \check{f}_j$ are polynomials over $\mathbb{F}_p$ and $\deg \hat{f}_j \leq \mathsf{S}_p^+(\omega)$, $\deg \check{f}_j \leq \mathsf{S}_p^-(\omega)$ for $j = 1,\dots,n$. Moreover, the polynomials $\hat{f}_j, \check{f}_j$ for $1 \leq j \leq n$ can be computed with $\mathcal{O}\big(n\big(\log(p)\,\mathsf{M}(n) + \mathsf{M}(ne)\big)\big)$ operations in $\mathbb{F}_p$.

Proof. Let $\omega = \sum_{i=0}^n w_i\, p^i$ be the signed p-ary expansion of ω with $|\omega_i| < p/2$. Then ψ_ω is represented as

$$\big(y+\zeta_\tau\big)^\omega = \frac{\prod_{w_i>0}\big(y+\zeta_\tau^{p^i}\big)^{w_i}}{\prod_{w_i<0}\big(y+\zeta_\tau^{p^i}\big)^{-w_i}}.$$

After adopting the notation

$$\hat{F}(y) = \prod_{w_i>0} \left(y + \zeta_\tau^{p^i}\right)^{w_i}, \quad \check{F}(y) = \prod_{w_i<0} \left(y + \zeta_\tau^{p^i}\right)^{|w_i|},$$

and $e^+ = \mathsf{S}_p^+(\omega)$, $e^- = \mathsf{S}_p^-(\omega)$, and writing

$$\hat{F}(y) = \sum_{i=0}^{e^+} \hat{\theta}_i\, y^i, \quad \check{F}(y) = \sum_{i=0}^{e^-} \check{\theta}_i\, y^i$$

and

$$\hat{\theta}_i = \sum_{j=1}^{n} \hat{c}_{i,j}\, \mathbf{x}_j, \quad \check{\theta}_i = \sum_{j=1}^{n} \check{c}_{i,j}\, \mathbf{x}_j$$

for some $\hat{c}_{i,j}$ and $\check{c}_{i,j}$ in $\mathbb{F}_p$, one can get

$$\hat{f}_j(y) = \sum_{i=0}^{e^+} \hat{c}_{i,j}\, y^i \quad \text{and} \quad \check{f}_j(y) = \sum_{i=0}^{e^-} \check{c}_{i,j}\, y^i.$$

The detailed complexity analysis for computing $\hat{f}_j, \check{f}_j$ is explained in the following. For any $0 \le i \le n$ we have

$$F_i(y) := \left(y + \zeta_\tau^{p^i}\right)^{|w_i|} = \sum_{k_i=0}^{|w_i|} \binom{|w_i|}{k_i} \zeta_\tau^{p^i(|w_i|-k_i)}\, y^{k_i} \in \mathbb{F}_{p^n}[y]. \tag{3.2}$$

Note that $\binom{|w_i|}{0}, \ldots, \binom{|w_i|}{|w_i|}$ are computed with $|w_i|$ multiplications and $|w_i|$ divisions in $\mathbb{F}_p$ and $\zeta_\tau^{p^i}, \ldots, \zeta_\tau^{p^i|w_i|}$ are computed with 1 exponentiation and $|w_i|$ multiplications in $\mathbb{F}_{p^n}$. Thus, all polynomials $F_0, \ldots, F_n \in \mathbb{F}_{p^n}[y]$ can be computed with $(n+1)\sum_{i=0}^{n}|w_i|$ multiplications and $(n+1)\sum_{i=0}^{n}|w_i|$ divisions in $\mathbb{F}_p$, along with $\mathcal{O}(n\log(p)) + \sum_{i=0}^{n}|w_i|$ multiplications in $\mathbb{F}_{p^n}$. Converting a multiplication in $\mathbb{F}_{p^n}$ into $\mathsf{M}(n)$ operations in $\mathbb{F}_p$, the computation of all F_i takes $\mathcal{O}\big(ne + (n\log(p)+e)\,\mathsf{M}(n)\big)$ operations in $\mathbb{F}_p$.

Once we have obtained F_i for all $0 \le i \le n$, two polynomials $\hat{F}, \check{F}$ in $\mathbb{F}_{p^n}[y]$ can be computed with $\mathcal{O}\big(n\,\mathsf{M}(ne)\big)$ operations in $\mathbb{F}_p$. Thus, the overall computational complexity is

$$\mathcal{O}\big(ne + (n\log(p)+e)\,\mathsf{M}(n) + n\,\mathsf{M}(ne)\big) \tag{3.3}$$

operations in $\mathbb{F}_p$. Since $ne = \mathcal{O}\big(\mathsf{M}(ne)\big)$ and $e\,\mathsf{M}(n) \le \mathsf{M}(ne)$, our claim is now evident from (3.3). □

3.2. **Algorithm description.** We present our simplification of Satoh's algorithm solving the SDL problem. Note that we extend the parameter choice by considering divisors of p^n-1, including divisors of $\Phi_n(p)$, which are the original choice of Satoh.

Theorem 3.2. *Let $\mathbb{G}$ be an abelian group of prime order p with generator g. Suppose that a generator ζ of the multiplicative group $\mathbb{F}_{p^n}^\times$ and a positive divisor D of p^n-1 are given.*[1] *Let r be an integer with $\gcd(D,r)=1$ and let $e = \mathsf{S}_p\Big(r\cdot\frac{p^n-1}{D}\Big)$. If g, g^{α^i} for $i = 1, \ldots, e$ are given, then α can be computed with $\mathcal{O}\big(n(e+n\,D^{1/2})\log(p)\big)$*

[1]Thoughout this paper, we assume that D does not divide p^m-1 for any proper divisor m of n otherwise it suffices to use the smaller m.

operations in $\mathbb{G}$ *and* $\mathcal{O}\big(n^2D^{1/2} + (n\log(p) + D^{1/2})\,\mathsf{M}(n) + \log(e)\log(ep)\,\mathsf{M}(e) + n\,\mathsf{M}(ne)\big)$ *operations in* $\mathbb{F}_p$ *by using storage for* $\mathcal{O}(nD^{1/2})$ *elements of* $\mathbb{G}$.

Proof. Let $E = \frac{p^n-1}{D}$, let $\mathbb{H}$ be the subgroup of order D of $\mathbb{F}_{p^n}^{\times}$, and let ζ_τ denote an element of $\mathbb{F}_{p^n}$ which is not contained in any proper subfield of $\mathbb{F}_{p^n}$. We consider a function defined on $\mathbb{F}_p$ which maps $\alpha \in \mathbb{F}_p$ to $\overline{\beta} \in \mathbb{H}$ as follows:

$$\begin{array}{rccc} \psi_{rE} : \mathbb{F}_p & \longrightarrow & \mathbb{F}_{p^n} & \longrightarrow & \mathbb{H} \\ \alpha & \longmapsto \overline{\alpha} := (\alpha + \zeta_\tau) & \longmapsto \overline{\beta} := \overline{\alpha}^{\,rE}. & & \end{array}$$

By Lemma 3.1, $\overline{\beta}$ can be represented by polynomials $\hat{f}_j$ and $\check{f}_j$ in α with coefficients in $\mathbb{F}_p$:

$$\overline{\beta} = \frac{\hat{f}_1(\alpha)\,\mathbf{x}_1 + \cdots + \hat{f}_n(\alpha)\,\mathbf{x}_n}{\check{f}_1(\alpha)\,\mathbf{x}_1 + \cdots + \check{f}_n(\alpha)\,\mathbf{x}_n}.$$

On the other hand, since $\mathbb{H}$ is generated by $\hat{\zeta}$, which can be obtained by raising the E-th power to the primitive element ζ of $\mathbb{F}_{p^n}$, there exists an integer $z \in \{0, \ldots, D-1\}$ such that

$$\overline{\beta} = \hat{\zeta}^{\,z}.$$

As in the Baby-Step-Giant-Step (BSGS) algorithm, we can find such an integer z by checking the equality

$$\overline{\beta}\,\hat{\zeta}^{\,-mu} = \hat{\zeta}^{\,v}, \tag{3.4}$$

or equivalently,

$$\Big\{\hat{f}_1(\alpha)\,\mathbf{x}_1 + \cdots + \hat{f}_n(\alpha)\,\mathbf{x}_n\Big\} \cdot \hat{\zeta}^{\,-mu} = \Big\{\check{f}_1(\alpha)\,\mathbf{x}_1 + \cdots + \check{f}_n(\alpha)\,\mathbf{x}_n\Big\} \cdot \hat{\zeta}^{\,v},$$

for all nonnegative integers u, v less than or equal to $m := \lceil \sqrt{D}\,\rceil$. Introducing the notation

$$\begin{aligned} \sum_{j=1}^{n} \hat{g}_{u,j}(\alpha)\,\mathbf{x}_j &= \Big\{\sum_{i=1}^{n} \hat{f}_i(\alpha)\mathbf{x}_i\Big\} \cdot \hat{\zeta}^{\,-mu}, \\ \sum_{j=1}^{n} \check{g}_{v,j}(\alpha)\,\mathbf{x}_j &= \Big\{\sum_{i=1}^{n} \check{f}_i(\alpha)\mathbf{x}_i\Big\} \cdot \hat{\zeta}^{\,v}, \end{aligned}$$

the equation (3.4) is equivalent to

$$\hat{g}_{u,j}(\alpha) = \check{g}_{v,j}(\alpha) \text{ for all } 1 \le j \le n,$$

which can be checked by

$$g^{\,\hat{g}_{u,j}(\alpha)} = g^{\,\check{g}_{v,j}(\alpha)} \text{ for all } 1 \le j \le n,$$

using $g, \ldots, g^{\alpha^e}$, because $\deg(\hat{g}_{u,j}) \le e^+ := \mathsf{S}_p^+(rE)$ and $\deg(\check{g}_{v,j}) \le e^- := \mathsf{S}_p^-(rE)$ for all $0 \le u, v \le m$ and all $1 \le j \le n$.

Thus, to find u and v, we first construct a lookup table with contains the tuples

$$\Big(g^{\,\hat{g}_{u,1}(\alpha)}, \ldots, g^{\,\hat{g}_{u,n}(\alpha)}\Big) \quad \text{for all} \quad 0 \le u \le m, \tag{3.5}$$

and then for each $0 \le v \le m$ we compute $\big(g^{\check{g}_{v,1}(\alpha)}, \ldots, g^{\check{g}_{v,n}(\alpha)}\big)$ and refer to the lookup table in order to find the unique pair (u, v) satisfying (3.4).

Once we find u, v satisfying equation (3.4), the number α is a common root of n polynomials $\hat{g}_{u,1} - \check{g}_{v,1}, \ldots, \hat{g}_{u,n} - \check{g}_{v,n}$ over $\mathbb{F}_p$. Thus, α can be computed

by finding the unique root satisfying the equation $g^x = g^\alpha$ among roots of the polynomial $\gcd(\hat{g}_{u,1} - \check{g}_{v,1}, \ldots, \hat{g}_{u,n} - \check{g}_{v,n})$.

The detailed complexity analysis is given in the following.

Compute the polynomials $\hat{f}_i$ and $\check{f}_i$ over $\mathbb{F}_p$. By Lemma 3.1, the polynomials $\hat{f}_i$ and $\check{f}_i$ for all $1 \le i \le n$ can be computed with $\mathcal{O}\big(n(\log(p)\,\mathsf{M}(n) + \mathsf{M}(ne))\big)$ operations in $\mathbb{F}_p$.

Compute the elements $g^{\hat{f}_i(\alpha)}$ and $g^{\check{f}_i(\alpha)}$ in $\mathbb{G}$. Since we know all coefficients of $\hat{f}_i$, $\check{f}_i$ and $\deg \hat{f}_i \le e^+$, $\deg \check{f}_i \le e^-$, we can compute $g^{\hat{f}_i(\alpha)}$ and $g^{\check{f}_i(\alpha)}$ for all $1 \le i \le n$ from $g, g^\alpha, \ldots, g^{\alpha^e}$. This step requires at most $2ne$ exponentiations in $\mathbb{G}$.

Compute the elements $\hat{\zeta}^{-mu}$ and $\hat{\zeta}^{v}$ in $\mathbb{F}_{p^n}$. The elements $\hat{\zeta} = \zeta^E$ and $\hat{\zeta}^{-m}$ are computed with two exponentiations in $\mathbb{F}_{p^n}$, which require $\mathcal{O}(n \log(p)\,\mathsf{M}(n))$ operations in $\mathbb{F}_p$. In addition $\hat{\zeta}^{-mu}$ and $\hat{\zeta}^{v}$ for all $0 \le u, v \le m$ are computed with $2m$ multiplications in $\mathbb{F}_{p^n}$, which require $\mathcal{O}(m\,\mathsf{M}(n))$ operations in $\mathbb{F}_p$. Thus, this step requires $\mathcal{O}\big((n\log(p) + D^{1/2})\,\mathsf{M}(n)\big)$ operations in $\mathbb{F}_p$.

Compute the elements $g^{\hat{g}_{u,j}(\alpha)}$ and $g^{\check{g}_{v,j}(\alpha)}$ in $\mathbb{G}$. We first compute $\hat{a}_j^{(u,i)} \in \mathbb{F}_p$ such that $\mathbf{x}_i\,\hat{\zeta}^{-mu} = \sum_{j=1}^n \hat{a}_j^{(u,i)} \mathbf{x}_j$. From the equation

$$\Big\{ \sum_{i=1}^n \hat{f}_i(\alpha)\,\mathbf{x}_i \Big\} \cdot \hat{\zeta}^{-mu} = \sum_{j=1}^n \Big\{ \sum_{i=1}^n \hat{a}_j^{(u,i)} \hat{f}_i(\alpha) \Big\} \mathbf{x}_j\,, \tag{3.6}$$

we have $\hat{g}_{u,j}(\alpha) = \sum_{i=1}^n \hat{a}_j^{(u,i)} \hat{f}_i(\alpha)$ and thus

$$g^{\hat{g}_{u,j}(\alpha)} = \prod_{i=1}^n \Big(g^{\hat{f}_i(\alpha)}\Big)^{\hat{a}_j^{(u,i)}}$$

is computed with n exponentiations in $\mathbb{G}$ using $g^{\hat{f}_i(\alpha)}$. Since the multiplication by $\mathbf{x}$ in $\mathbb{F}_{p^n}$ takes $\mathcal{O}(n)$ operations in $\mathbb{F}_p$, we can compute $g^{\hat{g}_{u,1}(\alpha)}, \ldots, g^{\hat{g}_{u,n}(\alpha)}$ with $\mathcal{O}(n^2 \log(p))$ operations in $\mathbb{G}$ and $\mathcal{O}(n^2)$ operations in $\mathbb{F}_p$. Similarly, we can compute $g^{\check{g}_{v,1}(\alpha)}, \ldots, g^{\check{g}_{v,n}(\alpha)}$ with the same complexity.

Recover α from the polynomials $\hat{g}_{u,1} - \check{g}_{v,1}, \ldots, \hat{g}_{u,n} - \check{g}_{v,n}$ over $\mathbb{F}_p$. If we know the numbers u, v satisfying equation (3.4), the polynomials $\hat{g}_{u,i}$, $\check{g}_{v,i}$ for all $1 \le i \le n$ can be computed with $\mathcal{O}(n^2 e)$ operations in $\mathbb{F}_p$ and two exponentiations in $\mathbb{F}_{p^n}$, because

$$\hat{g}_{u,j} = \sum_{i=1}^n \hat{a}_j^{(u,i)} \hat{f}_i \quad \text{and} \quad \check{g}_{v,j} = \sum_{i=1}^n \check{a}_j^{(v,i)} \check{f}_i,$$

where $\mathbf{x}_i\,\hat{\zeta}^{-mu} = \sum_{j=1}^n \hat{a}_j^{(u,i)} \mathbf{x}_j$, $\hat{a}_j^{(u,i)} \in \mathbb{F}_p$ and $\mathbf{x}_i\,\hat{\zeta}^{v} = \sum_{j=1}^n \check{a}_j^{(v,i)} \mathbf{x}_j$, $\check{a}_j^{(v,i)} \in \mathbb{F}_p$.

We note that the Euclidean algorithms for polynomials of degree at most ℓ over $\mathbb{F}_p$ can be implemented so as to take $\mathcal{O}(\mathsf{M}(\ell)\log(\ell))$ operations in $\mathbb{F}_p$ [25, Theorem 11.5] and all roots of the polynomial of degree ℓ in $\mathbb{F}_p$ can be found using an expected number of $\mathcal{O}(\mathsf{M}(\ell)\log(\ell)\log(\ell p))$ operations in $\mathbb{F}_p$ [25, Corollary 14.16].

Thus, α can be found from the polynomials $\hat{g}_{u,1} - \check{g}_{v,1}, \ldots, \hat{g}_{u,n} - \check{g}_{v,n}$ with at most e exponentiations in $\mathbb{G}$ and an expected number of $\mathcal{O}(\mathsf{M}(e)\log(e)\log(ep))$ operations in $\mathbb{F}_p$.

The storage requirement is $\mathcal{O}(nD^{1/2})$ elements in $\mathbb{G}$ for constructing the lookup table described in (3.5). Finally, assuming that the comparison cost to determine if

two elements of $\mathbb{G}$ are identical is faster than the group operation in $\mathbb{G}$, we achieve the overall complexity as claimed. □

Remark 3.3. If e is sufficiently large, then we can take $\mathsf{M}(e) = \mathcal{O}\big(e\log(e)\log(\log(e))\big)$ and $\mathsf{M}(ne) = \mathcal{O}\big(ne\log(ne)\log(\log(ne))\big)$. In this case, since $\mathsf{M}(n) = \mathcal{O}(n^2)$, the computational complexity in Theorem 3.2 becomes $\mathcal{O}\big(n(e + n\,D^{1/2})\log(p)\big)$ operations in $\mathbb{G}$ and $\widetilde{\mathcal{O}}\big(n^2(D^{1/2} + e + n\log(p)) + e\log(p)\big)$ operations in $\mathbb{F}_p$, where $\widetilde{\mathcal{O}}(\ell)$ denotes $\mathcal{O}(\ell\log^k(\ell))$ for some constant k.

4. Analysis

In this section we analyze the practicality of Satoh's generalization for solving the SDL problem, including our simplification.

The complexity presented in Theorem 3.2 is closely related to the choice of r, which shows that to solve the SDL problem more efficiently, one is required to find a good r such that $\mathsf{S}_p\big(r\cdot\frac{p^n-1}{D}\big)$ is as small as possible, before applying the proposed algorithm. One plausible way is to use a well-known lattice technique. Let $E = (p^n-1)/D$ and $\mathcal{L}$ be a n-dimensional lattice generated by $\big\{(E,0,\dots,0),$ $(-p,1,0,\dots,0),\ (-p^2,0,1,0,\dots,0),\ \dots,\ (-p^{n-1},0,\dots,0,1)\big\}$. Any vector $\mathbf{v} = (v_0,\dots,v_{n-1})$ in the lattice $\mathcal{L}$ satisfies $rE = v_0 + v_1p + \cdots + v_{n-1}p^{n-1}$ for some integer r. Since the volume of the lattice $\mathcal{L}$ is E, by the Minkowiski inequality on lattice [17], there is at least one vector $\mathbf{v} = (v_0,\dots,v_{n-1})$ in the lattice $\mathcal{L}$ with $\max_i|v_i| \le E^{1/n}$. If n is small, we may be able to compute this vector ν and the correponding integer r by using a well-known lattice reduction algorithm, such as LLL [14]. However, this approach does not tell us much about the lower bound on the length of the shortest vector in the lattice $\mathcal{L}$, and so is not sufficient to give a rigorous analysis on our algorithm, including Satoh's.

In the following subsection we explain how to resolve this problem in a more general setting, by analyzing the sum of digits $\sum_i|s_i|$ of any integers $s = \sum_{i\ge0} s_i\,t^i$, not just for the length of the shortest vector of the lattice $\mathcal{L}$, and give a lower bound on the number $\mathsf{S}_p\big(r\cdot\frac{p^n-1}{D}\big)$ for any r with $\gcd(D,r) = 1$.

4.1. Lower bound on sum of digits.

Let $m \ge 2$ be an integer. Let ξ denote a primitive m-th root of unity in $\overline{\mathbb{Q}}$ and consider the cyclotomic number field $K = \mathbb{Q}(\xi)$. The ring of integers of K is $\mathbb{Z}[\xi]$. Let $\mathfrak{a}$ be an ideal generated by two elements s and $(\xi - t)$. We then have the lemma about the norm of the ideal $\mathfrak{a}$.

Lemma 4.1. *Let $s,t \in \mathbb{Z}$ and $\mathfrak{a} = s\,\mathbb{Z}[\xi] + (\xi - t)\,\mathbb{Z}[\xi]$ be an ideal of $\mathbb{Z}[\xi]$. The norm of $\mathfrak{a}$ is*

$$N_{K/\mathbb{Q}}(\mathfrak{a}) = \big|\gcd\big(s,\Phi_m(t)\big)\big|.$$

Proof. Let $I = s\,\mathbb{Z}[\xi]$ and $J = (\xi - t)\,\mathbb{Z}[\xi]$. Then $N_{K/\mathbb{Q}}(I) = |N_{K/\mathbb{Q}}(s)| = s$, and $N_{K/\mathbb{Q}}(J) = |N_{K/\mathbb{Q}}(\xi - t)| = \Phi_m(t)$ because the minimal polynomial of $\xi - t$ is $\Phi_m(x + t)$. From $I, J \subseteq \mathfrak{a}$, we have $N_{K/\mathbb{Q}}(\mathfrak{a})\,|\,\gcd(s,\Phi_m(t))$. We claim that $\mathfrak{a}, 1+\mathfrak{a},\dots,(\gcd(s,\Phi_m(t))-1)+\mathfrak{a}$ are all the distinct elements of $\mathbb{Z}[\xi]/\mathfrak{a}$. Note that the claim is true if the following statement holds: for any integer c

$$c \equiv 0 \mod \mathfrak{a} \ \text{ is equivalent to } \ c \equiv 0 \mod \gcd(s,\Phi_m(t)). \tag{4.1}$$

To prove (4.1), we first show that $J \cap \mathbb{Z} = \Phi_m(t)\,\mathbb{Z}$. Clearly, $\Phi_m(t)\,\mathbb{Z} \subseteq J$, since $\Phi_m(t) \equiv \Phi_m(\xi) \equiv 0 \bmod J$. Let $\mathbf{y} = (\xi - t)Y(\xi) \in J$ for some $Y(x) = \sum_{i=0}^{\varphi(m)-1} y_ix^i \in \mathbb{Z}[x]$. Assume $\mathbf{y} \in \mathbb{Z}$. Define $\hat{Y}(x) = (x - t)Y(x) - y_{\varphi(m)-1}\Phi_m(x)$.

Then $\hat{Y}(x)-\mathbf{y}$ is an integer; otherwise, $1 \le \deg(\hat{Y}(x)-\mathbf{y}) < \varphi(m)$ and $\hat{Y}(\xi)-\mathbf{y}=0$, which contradicts the fact that the minimal polynomial of ξ is $\Phi_m(x)$. Moreover, $\hat{Y}(x)-\mathbf{y}=0$ since $\hat{Y}(\xi)-\mathbf{y}=0$. Evaluating $\hat{Y}(x)-\mathbf{y}$ at $x=t$, we have $\mathbf{y} = -y_{\varphi(m)-1}\Phi_m(t) \in \Phi_m(t)\,\mathbb{Z}$.

We proceed to prove (4.1). Let c be an integer with $c \equiv 0 \bmod \mathfrak{a}$. Then $c = sA(\xi) + (\xi - t)B(\xi)$ for some $A, B \in \mathbb{Z}[x]$. Since $A(x) = (x-t)Q(x) + A(t)$ for some $Q \in \mathbb{Z}[x]$, we have $c = sa + (\xi - t)C(\xi)$ for some $a \in \mathbb{Z}$ and $C \in \mathbb{Z}[x]$. From $c - sa = (\xi - t)C(\xi) \in J \cap \mathbb{Z} = \Phi_m(t)\,\mathbb{Z}$, we have $c \equiv 0 \bmod \gcd(s, \Phi_m(t))$. On the other hand, if c is an integer with $c \equiv 0 \bmod \gcd(s, \Phi_m(t))$, then $c = us + v\Phi_m(t)$ for some $u, v \in \mathbb{Z}$. Since $s \equiv 0 \bmod \mathfrak{a}$ and $\Phi_m(t) \equiv \Phi_m(\xi) \equiv 0 \bmod \mathfrak{a}$, we have $c \equiv 0 \bmod \mathfrak{a}$. □

From this lemma we can deduce a lower bound on the sum of digits. The following theorem is a generalization of lemma 3 in [12].

Theorem 4.2. *Let $m \ge 2$ and t be positive integers, and let $s = \sum_{i\ge 0} s_i\, t^i$ with $s_i \in \mathbb{Z}$. If the polynomial $\mathbf{s}(x) = \sum_{i\ge 0} s_i x^i$ is not divisible by the cyclotomic polynomial $\Phi_m(x)$, then*

$$\sum_{i\ge 0} |s_i| \ge \big|\gcd\big(s, \Phi_m(t)\big)\big|^{1/\varphi(m)}. \tag{4.2}$$

Proof. Let $\mathfrak{a} = s\,\mathbb{Z}[\xi] + (\xi - t)\,\mathbb{Z}[\xi]$ be an ideal of $\mathbb{Z}[\xi]$. Since $\xi \equiv t \bmod \mathfrak{a}$, $\mathbf{s}(\xi) \equiv 0 \bmod \mathfrak{a}$. Moreover, we have $\mathbf{s}(\xi) \in \mathfrak{a} \setminus \{0\}$, because $\mathbf{s}(x) \not\equiv 0 \bmod \Phi_m(x)$ implies (in fact, is equivalent to) $\mathbf{s}(\xi) \ne 0$. After noting that the norm of elements of $\mathfrak{a}$ is always divisible by the norm of $\mathfrak{a}$ and $N_{K/\mathbb{Q}}\big(\mathbf{s}(\xi)\big) \ne 0$, from Lemma 4.1, we have

$$\big|N_{K/\mathbb{Q}}\big(\mathbf{s}(\xi)\big)\big| \ge N_{K/\mathbb{Q}}(\mathfrak{a}) = \big|\gcd\big(s, \Phi_m(t)\big)\big|.$$

Let $\xi_1 = \xi, \ldots, \xi_{\varphi(m)}$ be all the conjugates of ξ over $\mathbb{Q}$. Observing that

$$N_{K/\mathbb{Q}}\big(\mathbf{s}(\xi)\big) = \prod_{j=1}^{\varphi(m)} \mathbf{s}(\xi_j) \quad \text{and} \quad |\mathbf{s}(\xi_j)| \le \sum_{i\ge 0} |s_i||\xi_j|^i \quad \text{and} \quad |\xi_j| = 1 \text{ for all } j,$$

we arrive at our claim:

$$\Big(\sum_{i\ge 0} |s_i|\Big)^{\varphi(m)} \ge \big|N_{K/\mathbb{Q}}\big(\mathbf{s}(\xi)\big)\big| \ge \big|\gcd\big(s, \Phi_m(t)\big)\big|.$$

□

Finally, combining this theorem with the inequality $\mathsf{S}_t(s) \ge \frac{1}{2}\sum_{i\ge 0} |s_i|$, we have a lower bound on $\mathsf{S}_t(\cdot)$.

Corollary 4.3. *Let t, $m \ge 2$ be integers and let $s = \sum_{i\ge 0} s_i\, t^i$ be an integer represented as the signed t-ary expansion. If $\sum_{i\ge 0} s_i x^i \not\equiv 0 \bmod \Phi_m(x)$, then*

$$\mathsf{S}_t(s) \ge \frac{1}{2}\big|\gcd\big(s, \Phi_m(t)\big)\big|^{1/\varphi(m)}. \tag{4.3}$$

4.2. **Analysis of algorithms.** We are now ready to explain why Satoh's generalized algorithm, including our simplification, are not efficient as ordinary DL-solving algorithms. Before continuing, we state the following lemma which will be used to verify our result.

Lemma 4.4 ([19, Theorem 94])**.** *If q is a prime which does not divide m, we have:*

(1) *The necessary and sufficient condition for the congruence $\Phi_m(x) \equiv 0 \bmod q$ to be solvable is that $q \equiv 1 \bmod m$.*
(2) *If $q \equiv 1 \bmod m$, the solutions of the congruence $\Phi_m(x) \equiv 0 \bmod q$ are the numbers whose order modulo q is m.*

4.2.1. *Our simplification.* Let D be a divisor of $p^n - 1$, $E = (p^n - 1)/D$, and r be an integer which is relatively prime to D. Let e denote the number $\mathsf{S}_p(rE)$.

We first take a close look at the complexity of our simplification. We claim that the overall running time of our simplification is at least

$$e + D^{1/2}.$$

The first term e comes from the complexity of computing the implicit representation of $\psi_{rE}(\alpha)$ of Section 3, which is almost equal to the degree of the polynomials $\hat{f}, \check{f}$, which, in turn, is equal to the number $\mathsf{S}_p(rE)$. The second term $D^{1/2}$ comes from the complexity of solving the DL problem over an auxiliary group $\mathbb{H}$ of order D, only given an implicit representation of the element of Section 3. One might be able to replace the BSGS algorithm by other randomized DL-solving algorithms, such as the Pollard rho or kangaroo method [21]. But, even if such a replacement is possible, the second term is not changed, because the expected running time of such randomized algorithms is $D^{1/2} + \mathcal{O}(1)$

Note that the ordinary DL-solving algorithms, such as Pollard's rho method, compute α in time $p^{1/2} + \mathcal{O}(1)$ only using two elements g and g^α. Thus, we are interested in the possibility that the complexity of our simplification is less than $p^{1/2}$. That is to say, we want to know if there exist parameters p, n, D, r satisfying the following four conditions, simultaneously:

$$D \,|\, p^n - 1\,, \quad D < p\,, \quad \mathsf{S}_p\Big(r \cdot \frac{p^n - 1}{D}\Big) < p^{1/2}\,, \quad \gcd(D, r) = 1.$$

The following proposition gives a negative answer to this question.

Theorem 4.5. *Let p be a prime, $n \geq 3$ be an integer, and let D be a divisor of $p^n - 1$. If D has a prime divisor q such that $q \nmid n$ and that $q \mid \Phi_m(p)$ for some divisor $m \geq 3$ of n, the complexity of the algorithm described in Theorem* 3.2 *is always greater than or equal to $p^{1/2}$.*

Proof. Let r be a positive integer with $\gcd(D, r) = 1$, and let $E = (p^n - 1)/D$ and $e = \mathsf{S}_p(rE)$. If d is greater than p, the complexity of our algorithm is greater than $p^{1/2}$, and thus we may assume that $D = p^\epsilon$ with $0 < \epsilon \leq 1$.

Since q is not a divisor of n and $\Phi_m(p) \equiv 0 \bmod q$, by Lemma 4.4 the order of p modulo q is m, which implies that

$$\Phi_\ell(p) \not\equiv 0 \bmod q \text{ for any divisor } \ell \neq m \text{ of } n.$$

We then have

$$E = \frac{p^n - 1}{D} = \frac{\prod_{\ell | n, \ell \neq m} \Phi_\ell(p)}{D/q} \times \frac{\Phi_m(p)}{q} \not\equiv 0 \bmod \Phi_m(p)\,.$$

Combining this with the fact $\gcd(D, r) = 1$, we have $rE \not\equiv 0 \bmod \Phi_m(p)$. Thus, by Corollary 4.3 we have

$$\begin{aligned} e = \mathsf{S}_p(rE) &\geq \frac{1}{2}\gcd\big(rE, \Phi_m(p)\big)^{1/\varphi(m)} \\ &\geq \frac{1}{2}\gcd\big(\ E\ , \Phi_m(p)\big)^{1/\varphi(m)} \\ &\geq \frac{1}{2}\left(\frac{\Phi_m(p)}{D}\right)^{1/\varphi(m)} \approx p^{1-\epsilon/\varphi(m)}. \end{aligned}$$

Since $\varphi(m) \geq 2$ and $0 < \epsilon \leq 1$, we have $0 < \epsilon/\varphi(m) \leq 1/2$. Thus, we arrive at our claim as follows:

$$e + D^{1/2} \geq p^{1-\epsilon/\varphi(m)} + p^{\epsilon/2} \geq p^{1/2}.$$

□

This theorem shows that if $n \geq 3$ the simplified algorithm is never better than the ordinary DL-solving algorithms except in the case where every prime divisor of D is a divisor of n or $\prod_{\ell|n,\ell\leq 2} \Phi_\ell(p)$. We note that in fact what we need to verify inefficiency of the simplified algorithm is the existence of a divisor $m \geq 3$ of n such that $\frac{p^n-1}{D} \not\equiv 0 \bmod \Phi_m(p)$. For example, letting n be even, $d_1 = \gcd(p^2 - 1, D)$, and $D_2 = D/D_1$, if D_2 is not a divisor of the greatest common divisor of all $\Phi_\ell(p)$ with $\ell \geq 3$ and $\ell \,|\, n$, we have such an m and thus the simplified algorithm fails to achieve improvement on the ordinary DL-solving algorithms.

4.2.2. *Satoh's algorithm.* We can apply similar arguments as just before to analyze the practicality of Satoh's generalization. We first summarize Satoh's result that appeared in [22].

Let $n \geq 2$ be an integer and let D be a proper divisor of $\Phi_n(p)$. The nonnegative integers $u, \Delta,$ and δ satisfy

$$u \cdot \frac{p^n - 1}{D} \equiv \Delta - \delta \bmod (p^n - 1)\,, \tag{4.4}$$

$$\gcd(u,\, p^n - 1) = 1, \quad 0 < u < p^{n-1}, \quad 0 \leq \Delta, \delta < p^n. \tag{4.5}$$

The overall complexity of Satoh's generalization is

$$\widetilde{\mathcal{O}}\big(n^2(n \log p + N + n^3 + D^{1/2}\,)\big),$$

where $N = \max\{||\Delta||_p, ||\delta||_p\}$ and $||\nu||_p$ denote $\sum_{i\geq 0} \nu_i$ for $\nu = \sum_{i\geq 0} \nu_i p^i$ with $0 \leq \nu_i < p$, which is the unsigend p-ary expansion of ν.

Although Satoh gave an upper bound on his algorithm, it takes at least

$$N + D^{1/2} \tag{4.6}$$

operations by the same reason as our simplification, where N is from the complexity computing an implicit representation of some element in $\mathrm{GL}_n(\mathbb{F}_p)$ and $D^{1/2}$ is the complexity of a BSGS-like procedure.

We now show that, for $n \geq 3$, (4.6) is always greater than or equal to $p^{1/2}$ for all parameters satisfying (4.4) and (4.5). As before, we may assume that D is a divisor of $\Phi_n(p)$ less than p, but is not a divisor of $\Phi_\ell(p)$ for any proper divisor ℓ of n, and $\Delta - \delta \geq 0$. Let $D = p^\epsilon$ for some $0 < \epsilon \leq 1$. Writing $u = qD + r$ for some integers q and $0 \leq r < D$, we have

$$r \cdot \frac{p^n - 1}{D} = \Delta - \delta \quad \text{and} \quad \gcd(D, r) = 1.$$

Let $E = (p^n - 1)/D$. By the assumption on D and the fact that $\gcd(D, r) = 1$, we have $rE \not\equiv 0 \bmod \Phi_n(p)$. This implies, by Corollary 4.3,

$$\mathsf{S}_p^+(rE) + \mathsf{S}_p^-(rE) \geq \gcd\big(rE, \Phi_n(p)\big)^{1/\varphi(n)}.$$

On the other hand, from the proposition 1 in [22], we have

$$N = \max\left\{||\Delta||_p, ||\delta||_p\right\} \geq \frac{1}{2} \cdot \Big\{ \mathsf{S}_p^+(rE) + \mathsf{S}_p^-(rE) \Big\}.$$

Combining these two inequalities, we have

$$N \geq \frac{1}{2} \gcd\left(rE, \Phi_n(p)\right)^{1/\varphi(n)} \geq \frac{1}{2} \left(\frac{\Phi_n(p)}{D} \right)^{1/\varphi(n)}.$$

Finally, since $\Phi_n(p) \approx p^{\varphi(n)}$ and $\varphi(n) \geq 2$, we may conclude that

$$N + D^{1/2} \geq p^{1-\epsilon/\varphi(n)} + p^{\epsilon/2} \geq p^{1/2}.$$

This shows that (4.6) is greater than or equal to $p^{1/2}$.

5. Conclusion and further studies

We studied Satoh's generalization of the $p + 1$ algorithm for solving the SDL problem with auxiliary inputs. We showed that, for $n \geq 3$, the mapping $\mathbb{F}_p \to \mathbb{F}_{p^n},\ \alpha \mapsto (\alpha + \zeta_\tau)^{(p^n-1)/D}$ is not so good for solving the SDL problem. One of the main problems when using this mapping is the occurrence of high degree polynomials. A possible direction of research to overcome this problem is to use high degree output obtained from cryptographic protocols, such as $g^{\frac{1}{\alpha+m}} = g^{(\alpha+m)^{p-2}}$, which is an output (a signature) from the Boneh-Boyen signature scheme [2]. It is also open to generalize the $p \pm 1$ algorithms to other auxiliary groups, such as elliptic curves. But, it seems that a new idea is needed to achieve this.

References

[1] Dan Boneh and Xavier Boyen, *Efficient selective-ID secure identity-based encryption without random oracles*, Advances in cryptology—EUROCRYPT 2004, Lecture Notes in Comput. Sci., vol. 3027, Springer, Berlin, 2004, pp. 223–238, DOI 10.1007/978-3-540-24676-3_14. MR2153175 (2006i:94041)

[2] Dan Boneh and Xavier Boyen, *Short signatures without random oracles and the SDH assumption in bilinear groups*, J. Cryptology **21** (2008), no. 2, 149–177, DOI 10.1007/s00145-007-9005-7. MR2386625 (2010c:94052)

[3] D. Boneh, X. Boyen, and E. Goh, Hierarchical identity based encryption without constant size ciphertext, in *Proceedings of Eurocrypt 2005*, LNCS 3494, pp. 440-456, 2005.

[4] Dan Boneh, Xavier Boyen, and Hovav Shacham, *Short group signatures*, Advances in cryptology—CRYPTO 2004, Lecture Notes in Comput. Sci., vol. 3152, Springer, Berlin, 2004, pp. 41–55, DOI 10.1007/978-3-540-28628-8_3. MR2147494 (2006d:94075)

[5] D. R. L. Brown and R. P. Gallant, The static Diffie-Hellman problem, Cryptology ePrint Archive, Report no. 2004/306, 2004. Available from `http://eprint.iacr.org/2004/306`.

[6] Dan Boneh, Craig Gentry, and Brent Waters, *Collusion resistant broadcast encryption with short ciphertexts and private keys*, Advances in cryptology—CRYPTO 2005, Lecture Notes in Comput. Sci., vol. 3621, Springer, Berlin, 2005, pp. 258–275, DOI 10.1007/11535218_16. MR2237311 (2007a:94181)

[7] Xavier Boyen, *The uber-assumption family: a unified complexity framework for bilinear groups*, Pairing-based cryptography—Pairing 2008, Lecture Notes in Comput. Sci., vol. 5209, Springer, Berlin, 2008, pp. 39–56, DOI 10.1007/978-3-540-85538-5_3. MR2733903 (2012m:94255)

[8] Jung Hee Cheon, *Security analysis of the strong Diffie-Hellman problem*, Advances in cryptology—EUROCRYPT 2006, Lecture Notes in Comput. Sci., vol. 4004, Springer, Berlin, 2006, pp. 1–11, DOI 10.1007/11761679_1. MR2423212 (2009d:94073)
[9] Jung Hee Cheon, *Discrete logarithm problems with auxiliary inputs*, J. Cryptology **23** (2010), no. 3, 457–476, DOI 10.1007/s00145-009-9047-0. MR2643686 (2011i:94064)
[10] Bert den Boer, *Diffie-Hellman is as strong as discrete log for certain primes*, Advances in cryptology—CRYPTO '88 (Santa Barbara, CA, 1988), Lecture Notes in Comput. Sci., vol. 403, Springer, Berlin, 1990, pp. 530–539, DOI 10.1007/0-387-34799-2_38. MR1046405
[11] Whitfield Diffie and Martin E. Hellman, *New directions in cryptography*, IEEE Trans. Information Theory **IT-22** (1976), no. 6, 644–654. MR0437208 (55 #10141)
[12] Florian Hess, *Pairing lattices*, Pairing-based cryptography—Pairing 2008, Lecture Notes in Comput. Sci., vol. 5209, Springer, Berlin, 2008, pp. 18–38, DOI 10.1007/978-3-540-85538-5_2. MR2733902
[13] A. A. Karatsuba and Y. Ofman, Multiplication of multidigit numbers on automata, *Soviet Physics Doklady*, Vol. 7, pp. 595–596, 1963.
[14] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), no. 4, 515–534, DOI 10.1007/BF01457454. MR682664 (84a:12002)
[15] Ueli M. Maurer, *Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms*, Advances in cryptology—CRYPTO '94 (Santa Barbara, CA, 1994), Lecture Notes in Comput. Sci., vol. 839, Springer, Berlin, 1994, pp. 271–281, DOI 10.1007/3-540-48658-5_26. MR1316411 (95k:94021)
[16] Ueli M. Maurer and Stefan Wolf, *The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms*, SIAM J. Comput. **28** (1999), no. 5, 1689–1721, DOI 10.1137/S0097539796302749. MR1694168 (2000d:11154)
[17] H. Minkowski, Geometrie der Zahlen, Leipzig und Berlin, Druck und Verlag von B. G. Teubner, 1910.
[18] S. Mitsunari, R. Sakai, and M. Kasahara, A new traitor tracing, *IEICE Trans. Fundamentals*, Vol. E58-A, No. 2, pp. 481-484, 2002.
[19] Trygve Nagell, *Introduction to number theory*, Second edition, Chelsea Publishing Co., New York, 1964. MR0174513 (30 #4714)
[20] Tatsuaki Okamoto, *Efficient blind and partially blind signatures without random oracles*, Theory of cryptography, Lecture Notes in Comput. Sci., vol. 3876, Springer, Berlin, 2006, pp. 80–99, DOI 10.1007/11681878_5. MR2241667 (2007e:94089)
[21] J. M. Pollard, *Monte Carlo methods for index computation* (mod p), Math. Comp. **32** (1978), no. 143, 918–924. MR0491431 (58 #10684)
[22] T. Satoh, On generalization of Cheon's algorithms, Cryptology ePrint Archive, Report no. 2009/058, 2009. Available from `http://eprint.iacr.org/2009/058`.
[23] Daniel Shanks, *Class number, a theory of factorization, and genera*, 1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, N.Y., 1969), Amer. Math. Soc., Providence, R.I., 1971, pp. 415–440. MR0316385 (47 #4932)
[24] A. Schönhage and V. Strassen, *Schnelle Multiplikation grosser Zahlen* (German, with English summary), Computing (Arch. Elektron. Rechnen) **7** (1971), 281–292. MR0292344 (45 #1431)
[25] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, 2nd ed., Cambridge University Press, Cambridge, 2003. MR2001757 (2004g:68202)

The Attached Institute of ETRI, P.O. Box 1, Yuseong, Daejeon, 305-600, Korea
E-mail address: `mkkim@ensec.re.kr`

ISaC and Department of Mathematical Sciences, Seoul National University, Seoul 151-747, Korea
E-mail address: `jhcheon@snu.ac.kr`

ISaC and Department of Mathematical Sciences, Seoul National University, Seoul 151-747, Korea
E-mail address: `isll@snu.ac.kr`