On Generalization of Cheon's Algorithm

Takakazu Satoh[⋆]

Department of Mathematics,
Tokyo Institute of Technology, Tokyo, 152-8551, Japan
satohaar@mathpc-satoh.math.titech.ac.jp

Abstract. Let G be a cyclic group generated by g whose group operations are written additively. Assume that the order of G is a prime p. Let d be a divisor of $p+1$. Let $c \in \mathbf{F}_p$. Given cg, c^2g, …, $c^{2d}g$, Cheon[4] gave an algorithm to compute c more efficiently than solving ordinal discrete logarithm problems. With improvement by Kozaki, Kutsuma and Matsuo[5], the algorithm runs with $O(\max(d, \sqrt{p/d}))$ group operations. We generalize his algorithm for divisors of $\varphi_n(p)$ where $n \in \mathbf{N}$ and φ_n is the n-th cyclotomic polynomial. In case that d is a divisor of $p+1$ (i.e. the case $n=2$), our algorithm requires only cg, …, c^dg to compute c with $\tilde{O}(\max(d, \sqrt{p/d}))$ operations of G and $\mathbf{F}_p$.

Key words: discrete log problem, generic algorithm, Cheon's algorithm

1 Introduction

Let $G := \langle g \rangle$ be a cyclic group of prime order p. We write the group operations of G additively. The classical discrete logarithm problem (DLP) with respect to the base point g is a problem to compute $c \in \mathbf{F}_p$ from input date g and cg. We can compute c by using Shank's Baby Step Giant Step (BSGS) algorithm[8] with $O(\sqrt{p})$ group operations. It is independently proved by Nechaev[7] and Shoup[9] that the growth rate estimate is best possible for generic algorithms solving DLP. In Eurocrypt 2006, Cheon[4] made a breakthrough. Let d be a divisor of $p-1$. For given cg and c^dg, he gave an algorithm to compute c with $O(\log p \max(\sqrt{d}, \sqrt{p/d}))$ group operations. Kozaki, Kutsuma and Matsuo reduced a number of group operation to $O(\max(d, \sqrt{p/d}))$. The l-strong Diffie-Hellman problem is to compute $c^{l+1}g$ from given g, cg, …, c^lg. (See e.g., Boneh and Boyen[2], Boneh, Boyen and Goh[3].) One of important consequences of Cheon's algorithm is that the difficulty of the strong Diffie-Hellman problems depends on p even under generic algorithms.

Cheon[4] also gave an algorithm for a divisor d of $p+1$. For given cg, c^2g, …, $c^{2d}g$, Cheon's algorithm together with Kozaki et al.'s technique computes c with $O(\max(d, \sqrt{p/d}))$ group operations. This is based on a certain embedding from $\mathbf{F}_p^\times$ to the unique subgroup of $\mathbf{F}_{p^2}^\times$ of order $p+1$. In [4, Remark 1], Cheon wrote: (group operation notation is changed to additive notation)

[⋆] The work was supported by the Grant-in-Aid for the Scientific Research (B) 18340005.

> in our situations we can use the Diffie-Hellman oracle $DH(xg, yg) = xyg$ only when x is fixed and $y = x^l$ for some small l. This restriction is an obstacle when we try to generalize the proposed algorithm into other extension fields of $\mathbf{F}_p$ or elliptic or hyperelliptic curves over $\mathbf{F}_p$.

In this paper, we generalize Cheon's algorithm to the case that d satisfies the following two conditions.

(1) There exists an integer n for which $d|\varphi_n(p)$, where $\varphi_n(X)$ is the n-th cyclotomic polynomial.

(2) There exists integers u, $\mathit{\Delta}$ and δ such that $\frac{du(p^n-1)}{\varphi_n(p)} \equiv \mathit{\Delta} - \delta \bmod (p^n - 1)$ and that $0 \leq \mathit{\Delta}, \delta, u < p^n - 1$ and that $\gcd(p^n - 1, u) = 1$.

For an integer $a := \sum_{i\geq 0} a_i p^i$ with $0 \leq a_i < p$, we define its p-adic weight $\|a\|_p$ as $\sum_{i\geq 0} a_i$. Put $w := \|\mathit{\Delta}\|_p - \|\delta\|_p$. We show that, given $cg, \ldots, c^w g$, we can compute c with $\tilde{O}(n^2(n \log p + w + n^3 + \sqrt{D}))$ group operations and field arithmetic in $\mathbf{F}_p$. Cheon's algorithm for a divisor of $p+1$ corresponds to the case of $n = 2$ of our algorithm. However, in this case, there is a choice of u, $\mathit{\Delta}$ and δ which gives $w = d$ and our algorithm needs only $cg, \ldots, c^d g$ for input and $\tilde{O}(\max(d, \sqrt{p/d}))$ group operations and arithmetic operations of $\mathbf{F}_p$. An effectiveness of our algorithm for the cases $n \geq 3$ is not clear. Numerical experiments suggests that there exists an integer $n \leq 14$ and a divisor D of $\varphi_n(p)$ satisfying $D \approx p^{2/3}$ for most of p. Although $\varphi_n(p)$ is too large to factorize completely, finding its divisor of bit size 20∼60 (if any) is not hard if we use elliptic curve factorization method by Lenstra[6]. One more thing to be considered is that in pairing based cryptography, it is *necessary* to use use p such that (1) is fulfilled with small n. Thus, difficulty to find c depends on existence of u, $\mathit{\Delta}$ and δ satisfying (2) for which $\|\mathit{\Delta}\|_p - \|\delta\|_p$ is small. Although (2) is stated in the elementary number theory, finding such u, $\mathit{\Delta}$ and δ is much complicated than it looks. So far, the author obtained neither proof of non-existence of such u, $\mathit{\Delta}$ and δ for $n \geq 3$ nor non-trivial example of u, $\mathit{\Delta}$ and δ. Probably, it would be a prudent choice to use a larger p until we obtain more precise understanding on Cheon's algorithm.

The rest of the paper is organized as follows. In Section 2, we present our generalization of Cheon's algorithm. In Section 3, we show Cheon's algorithm is a special case of our algorithm. In Section 4, we present some results of numerical experiments and we discuss cryptographic implication of our results.

Notation.

Throughout the paper, the word "operation" means either arithmetic operations of $\mathbf{F}_p$ or group operations of G unless otherwise noted. We denote the group of n-th roots of unity by μ_n.

2 Generalized Cheon's Algorithm

In this section, we give our generalization of Cheon's algorithm. Let p be an odd (large) prime. Let $G = \langle g \rangle$ be a cyclic group of order p whose group operation is written additively. Our main idea is to embed G to $GL(n, \mathbf{F}_p)$ rather than a multiplicative group of an extension field of $\mathbf{F}_p$.

Given cg, c^2g, …, we want to compute $c \in \mathbf{F}_p$. Let $n \geq 2$ be an integer. We denote the n-th cyclotomic polynomial by φ_n. We extend Cheon's algorithm in the case that $\varphi_n(p)$ has a certain divisor whose explicit condition is given later.

Put $q := p^n$. Let $\theta \in \mathbf{F}_q^\times$ be a generator of $\mathbf{F}_q$ over $\mathbf{F}_p$ and let $\chi_\theta(T) := T^n + a_{n-1}T^{n-1} + \cdots + a_1 T + a_0$ be the monic minimal polynomial of θ over $\mathbf{F}_p$. Note $\mathbf{F}_p(\theta) = \mathbf{F}_q$. Let Θ be the representation matrix of the multiplication by θ on $\mathbf{F}_p(\theta)$ with respect to the base $\{ 1, \theta, \ldots, \theta^{n-1} \}$. Thus,

$$\Theta := \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 & -a_0 \\ 1 & 0 & & & \vdots & -a_1 \\ 0 & 1 & 0 & & \vdots & -a_2 \\ 0 & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -a_{n-1} \end{pmatrix}.$$

Note that the characteristic polynomial of Θ coincides with χ_θ. Denote the p-th power Frobenius map by π_p. Define a ring homomorphism ε from $\mathbf{F}_q$ to $\mathrm{Mat}(n, \mathbf{F}_q)$ by

$$\varepsilon(z) := \mathrm{diag}(z, \pi_p(z), \ldots, \pi_p^{n-1}(z))$$

where $\mathrm{diag}(a_1, \ldots, a_n)$ is the diagonal matrix whose i-th diagonal entry is a_i. Since χ_θ is irreducible over $\mathbf{F}_p$, all the roots of χ_θ are θ, $\pi_p(\theta)$, …, $\pi_p^{n-1}(\theta)$. Let $v_0 \in \mathbf{F}_q^n$ be an eigenvector of Θ for the eigenvalue z. Then, $\pi_p^i(v_0)$ is an eigenvector for the eigenvalue $\pi_p^i(\theta)$. On the other hand, χ_θ is square free and thus Θ is diagonisable. Specifically, we have

$$V^{-1}\Theta V = \varepsilon(\theta)$$

where $V := (v_0 \ \ \pi_p(v_0) \ \ \cdots \ \ \pi_p^{n-1}(v_0))$.

Lemma 1. *For any $z \in \mathbf{F}_q$, it holds that $V\varepsilon(z)V^{-1} \in \mathrm{Mat}(n, \mathbf{F}_p)$.*

Proof. There exist a_0, …, $a_{n-1} \in \mathbf{F}_p$ satisfying $z = \sum_{i=0}^{n-1} a_i \theta^i$. Then, $V\varepsilon(z)V^{-1} = V\varepsilon\left(\sum a_i\theta^i\right)V^{-1} = V\sum a_i \varepsilon(\theta)^i V^{-1} = \sum a_i \Theta^i \in \mathrm{Mat}(n, \mathbf{F}_p)$. □

Definition 1. *For a given integer ν, we put $\|\nu\|_p := \sum_{i\geq 0} \nu_i$ where ν_i is an integer satisfying $0 \leq \nu_i < p$ and $\sum_{i \geq 0} \nu_i p^i = \nu$.*

Lemma 2. *Let ν be a positive integer less than p^n and put $N := \|\nu\|_p$. Then there exists $N + 1$ matrices A_0, …, $A_N \in \mathrm{Mat}(n, \mathbf{F}_p)$ satisfying*

$$(xI + \Theta)^\nu = \sum_{j=0}^{N} x^j A_j \tag{3}$$

for all $x \in \mathbf{F}_p$. We can compute $A_0, \ldots, A_N$ with $\tilde{O}(n^3(\log p + N + n^2))$ operations.

Proof. Since $\nu < p^n$, we can write $\nu = \sum_{i=0}^{n-1} \nu_i p^i$ with $0 \leq \nu_i < p$. Observe that

$$(x+\theta)^\nu = \prod_{i=0}^{n-1} (x+\theta)^{\nu_i p^i} = \prod_{i=0}^{n-1} (x + \pi_p^i(\theta))^{\nu_i}.$$

for $x \in \mathbf{F}_p$. In practice n is not so large. So we use naive arithmetic algorithms to estimate the time complexity of arithmetic operations of $\mathbf{F}_q$ in terms of those of $\mathbf{F}_p$. We can compute $\pi_p(\theta), \ldots, \pi_p^{n-1}(\theta)$ with $O(n \log p)$ multiplications in $\mathbf{F}_q$ which amounts to $\tilde{O}(n^3 \log p)$ arithmetic operations in $\mathbf{F}_p$. (Whereas, we use an FFT based polynomial multiplication algorithm for multiplications of polynomials with coefficients in $\mathbf{F}_q$ since ν_i and N are as large as such an asymptotic algorithm is very efficient.) Let X be an indeterminate. We can compute $(X + \pi_p^i(\theta))^{\nu_i} \in \mathbf{F}_q[X]$ with $\tilde{O}(n^2 \nu_i)$ operations for each i. Computing products of these n polynomials with $\tilde{O}(n^3 N)$ operations, we obtain $a_0, \ldots, a_N \in \mathbf{F}_q$ satisfying

$$(x+\theta)^\nu = \sum_{j=0}^{N} x^j a_j. \tag{4}$$

Applying ε, we see that

$$(xI + \Theta)^\nu = V(xI + \varepsilon(\theta))^\nu V^{-1} = \sum_{j=0}^{N} x^j V \varepsilon(a_j) V^{-1}.$$

Thus we can take $V\varepsilon(a_i)V^{-1}$ as A_j satisfying (3) for all $x \in \mathbf{F}_p$. The assertion $A_j \in \mathrm{Mat}(n, \mathbf{F}_p)$ follows from Lemma 1. Note that we have already computed $\pi_p(\theta), \ldots, \pi_p^{n-1}(\theta)$. Hence we can compute Θ with $\tilde{O}(n^5)$ operations with naive matrix multiplication algorithm. Since $\Theta \in \mathrm{Mat}(n, \mathbf{F}_p)$, we obtain $\Theta^2, \ldots, \Theta^{n-1}$ with $O(n^4)$ operations Thus for each j we obtain A_j from a_j with $O(n^3)$ operations. These estimates give overall time complexity. □

Definition 2. *Given $\nu \in \mathbf{N}$, we denote the matrix coefficient polynomial $\sum_{j=0}^{N} X^j A_j$ by $R_\nu(X)$.*

Now we can state our generalization of Cheon's algorithm. Let d be a divisor of $\varphi_n(p)$ and put $D := \varphi_n(p)/d$. Take a generator θ of $\mathbf{F}_q$ over $\mathbf{F}_p$ satisfying

$$\theta^d \neq 1. \tag{5}$$

(Testing random elements of $\mathbf{F}_q^\times$ will give such a θ sooner or later.) Let $c \in \mathbf{F}_p$ and put $M := cI + \Theta$. Note $V^{-1}MV = \varepsilon(c+\theta)$. Let ζ be a primitive D-th root of unity. Since $(c+\theta)^{(p^n-1)/D} \in \mu_D$, there exists an integer m satisfying $0 \leq m < D$ and $(c+\theta)^{(p^n-1)/D} = \zeta^m$, or equivalently

$$M^{(p^n-1)/D} = \Omega^m \quad \text{where} \quad \Omega := V\varepsilon(\zeta)V^{-1}. \tag{6}$$

We note $\Omega \in \mathrm{Mat}(n, \mathbf{F}_p)$ by Lemma 1. Suppose we found non-negative integers u, Δ and δ satisfying

(7) $\gcd(u, p^n - 1) = 1$, $0 \leq \varDelta < p^n$, $0 \leq \delta < p^n$, $0 < u < p^{n-1}$.
(8) $u(p^n - 1)/D \equiv \varDelta - \delta \bmod (p^n - 1)$.

Note that (7) implies $\gcd(u, D) = 1$. Thus (6) holds if and only if

$$M^{\varDelta} = \varOmega^{mu} M^{\delta}. \tag{9}$$

We use Shanks' Baby Step Giant Step algorithm to find m. Put $L = \left\lceil \sqrt{D} \right\rceil$. There exists integers m_1 and m_2 satisfying $0 \leq m_1 < L$, $0 \leq m_2 < L$ and $m = m_1 L + m_2$. Multiplying $\varOmega^{-m_1 L}$ to (9) from left, we see that (9) is equivalent to

$$\varOmega^{-m_1 u L} M^{\varDelta} = \varOmega^{m_2 u} M^{\delta}. \tag{10}$$

Once we have obtained m, we compute c as a solution of

$$r_{\varDelta}(x) = \zeta^{mu} r_{\delta}(x) \tag{11}$$

where $r_\nu(x)$ is the right hand side of (4) for $\nu \in \mathbf{N}$. This is a polynomial equation with respect to x of degree at most $w := \max(\|\varDelta\|_p, \|\delta\|_p)$. This is obvious except possibly for the case $m = 0$ in which case (11) might be an identity. However, evaluating (11) at $x = 0$, we obtain $\theta^d = 1$, which contradicts to the choice of θ as (5). Therefore, (11) is not an identity. With asymptotic fast algorithms, solving (11) costs $\tilde{O}(w)$ operations.

It remains to see how we can test whether (10) holds or not. Since the order of G is p, the set G^n is a left $\mathrm{Mat}(n, \mathbf{F}_p)$ module by the natural action of $\mathbf{F}_p$ on G. Explicitly, the action is

$$(r_{ij})_{1 \leq i,j \leq n} (g_i)_{1 \leq i \leq n} := \left(\sum_{j=1}^{n} r_{ij} g_j \right)_{1 \leq i \leq n}.$$

Let $e_i \in G^n$ be a column vector whose i-th row is g and other rows are $0 \in G$.

Lemma 3. *Eq. (10) holds if and only if*

$$\varOmega^{-m_2 u L} R_{\varDelta}(c) e_1 = \varOmega^{m_1 u} R_{\delta}(c) e_1. \tag{12}$$

Proof. Note that $M^\nu = R_\nu(c)$ for any $c \in \mathbf{F}_p$ and $\nu \in \mathbf{N}$. Hence "only if" part is obvious. To prove converse, note that, as a left $\mathrm{Mat}(n, \mathbf{F}_p)$ module, G^n is isomorphic to $\mathbf{F}_p^n$. Hence it is enough to show

$$\varOmega^{-m_2 u L} R_{\varDelta}(c) e_i = \varOmega^{m_1 u} R_{\delta}(c) e_i \tag{13}$$

for all $1 \leq i \leq n$. By the definition of $\varTheta$, we see $e_i = \varTheta^{i-1} e_1$. However, $\varTheta$ commutes with M and $\varOmega$. Multiplying $\varTheta^{i-1}$ to (12) from left, we obtain (13). □

We end this section with the resulting algorithm.

Algorithm 1.
Input: p, n, χ_θ, d, u, $\varDelta$, δ, g, cg, c^2g, …, c^wg
Output: c
Procedure:

```
 1: $D := \varphi_n(p)/d$ ; $w := \|\varDelta\|_p - \|\delta\|_p$ ; $\zeta := \theta^{(p^n-1)/D}$ ;
 2: compute $\varTheta$, $R_\varDelta(X)$ and $R_\delta(X)$
 3: $v_1 := R_\varDelta(c)e_1$ ; $v_2 := R_\delta(c)e_1$ ;
 4: compute $\varOmega^u$
 5: $L := \left\lceil \sqrt{D} \right\rceil$ ;
 6: Build a look up table consisting of $(m_2, \varOmega^{m_2 u}v_2)$ for $m_2 = 0, \ldots, L$
 7: Find a match $\varOmega^{-m_1 u L}v_1 = \varOmega^{m_2 u}v_2$
 8: $m := m_1 L + m_2$
 9: find roots $\xi$ of $r_\varDelta(x) = \zeta^{mu} r_\delta(x)$ in $\mathbf{F}_p$
10: output $\xi$ if $\xi g = cg$
```

With the technique used in Kozaki, Kutsuma and Matsuo[5], we can perform Step 6 and Step 7 with $O(n^2\sqrt{D})$ operations. Thus overall time complexity is $\tilde{O}(n^2(n\log p + w + n^3 + \sqrt{D}))$ operations.

3 Relation to Cheon's Algorithm

In this section, we observe that Cheon's algorithm is the case $n = 2$ of our algorithm. We keep notation in the previous section.

Let d be a proper divisor of $p + 1 = \varphi_2(p)$. Then $D = (p + 1)/d$ and $(p^2 - 1)/D = (p - 1)d$. Let ζ_{p+1} be a (fixed) primitive $(p + 1)$-th root of unity and put $\zeta_D := \zeta_{p+1}^{(p+1)/D}$. We take $u = 1$, $\varDelta = pd$ and $\delta = d$. Then, $\|\varDelta\|_p = \|\delta\|_p = d$. Thus with d inputs cg, c^2g, …, c^dg, we can run Algorithm 1. In this case, as is Cheon's original algorithm, we can obtain c without solving a degree d equation. Suppose we have obtained m satisfying

$$(c+\theta)^{pd} = \zeta_D^m (c+\theta)^d. \tag{14}$$

In order to read off c, Cheon used the BSGS again, which requires only $O(\sqrt{d})$ operations. This is a result of the choice of $\varDelta$ and δ. By (14), there exists $k \in \mathbf{N}$ such that $(c+\theta)^{p-1} = \zeta_{p+1}^{m+\frac{p+1}{d}k}$. We can apply the BSGS procedure to find k. Put $n := m + \frac{p+1}{d}k$. Then (14) yields $c + \theta^p = \zeta_{p+1}^n(c+\theta)$. Hence, we obtain $c = \frac{\zeta_{p+1}^n\theta - \theta^p}{1-\zeta_{p+1}^n}$ with no more group operations.

Cheon's original algorithm requires $2d+1$ inputs g, cg, …, $c^{2d}g$. The reason why we need only $d+1$ inputs g, cg, …, c^dg comes from the decomposition of $(p+1)/d$ to a differences of low weight integers. In Cheon's original algorithm, a non-square element $t \in \mathbf{F}_p^\times$ and its square root $\psi \in \mathbf{F}_{p^2}^\times$ are used to construct

$$\frac{1+tc^2}{1-tc^2} + \frac{2c}{1-tc^2}\psi \in \mu_{p+1} \subset \mathbf{F}_{p^2}^\times$$

from an unknown $c \in \mathbf{F}_p^\times$. In our method $c+\theta \notin \mu_{p+1}$ in general but $(c+\theta)^{p-1} \in \mu_{p+1}$ and $p-1 = 1 \cdot p - 1$.

We note that the O-constant in the time complexity of Cheon's original algorithm is smaller than that of our algorithm. However, decreasing number of inputs seems to be more important in case that we obtain c^2g, c^3g, ... by communicating other entity who knows c. For example, Cheon[4, Sect. 4.1] proposed a "protocol" to obtain $c^{n+1}g$ from c^ng by requesting one blind signature due to Boldyreva[1]. Reducing number of signature queries from $2d$ to d makes an attempt to obtain necessary input data less conspicuous.

4 Cryptographic Implication of our Algorithm

In this section, we consider time complexity of our algorithm and its cryptographic implications. We keep the notation in the previous sections. Let $t \leq \frac{1}{2}$. The conditions on p in order that the time complexity of our algorithm is $\tilde{O}(p^t)$ is that there exist integers n, d, Δ, δ, u satisfying the following conditions:

(a) $d|\varphi_n(p)$ and $\frac{n^2\varphi_n(p)}{d} = \tilde{O}(p^{2t})$
(b) $\frac{ud(p^n-1)}{\varphi_n(p)} \equiv \Delta - \delta \bmod (p^n-1)$ where $0 \leq \Delta < p^n - 1$, $0 \leq \delta < p^n - 1$ and $\gcd(u, p^n-1) = 1$.
(c) $n^2w = \tilde{O}(p^t)$ where $w = \max(\|\Delta\|_p, \|\delta\|_p)$.

In order to perform arithmetic operation in $\mathbf{F}_q$, it seems adequate to determine an upper bound N of n. Although there is no theoretical support, the integer d satisfying (a) seems to exists for most of p unless N or t is too small. The author ran numerical experiments with $N = 14$. Among 18 randomly generated 140 bit primes p, only two primes do not have a divisor d of $\varphi_n(p)$ satisfying $2^{70} \leq d < 2^{105}$ and $3 \leq n \leq 14$. For 150 bit primes, among 23 randomly generated primes, only one prime does not have a divisor d of $\varphi_n(p)$ satisfying $2^{75} \leq d < 2^{113}$ and $3 \leq n \leq 14$. In order to find divisors, a ρ-method based factoring algorithm up to 2^{24} steps was used. Probably, with some increase of N, the condition (a) is always fulfilled for most of p.

Given d satisfying (a) and any integer u satisfying $\gcd(u, p^n-1) = 1$, there always exist many integers Δ and δ satisfying (b). Thus, difficulty of the w-strong Diffie-Hellman problem (or inefficiency of our algorithm) depends on the non-existence of integers u, Δ and δ satisfying (c).

For a particular u, the proposition given at the end of this section (where we take $\nu = \frac{ud(p^n-1)}{\varphi_n(p)}$) gives a criterion of existence of Δ and δ which gives small w. However, to the best knowledge of current author, there is no known criterion or algorithm to test existence of u, Δ and δ giving small w. Whether one can intentionally create p for which our algorithm especially runs faster.

Assuming $w \approx \sqrt{\varphi_n(p)/D}$ (which holds for Cheon's original algorithm), our algorithm is most efficient when $w \approx p^{1/3}$. However we cannot take w so large due to space complexity constraint. In practice, $w \approx 2^{20}$ seems to be well above feasible value. Under the assumption, $\sqrt{D}$, the search size for the BSGS process involved in our algorithm, is smaller by a factor of $\sqrt{w}$ than $\sqrt{p}$. This means that in order to compensate security loss, even if our algorithm works with best efficiency, we need to increase the bit size of p by the bit size of w.

We end this section with a proof of an inequality used in the above argument.

Proposition 1. *Let $\nu = \sum_{i \geq 0} a_i p^i$ with $|a_i| < \frac{p}{2}$ be the signed p-adic expansion of ν and put $\nu_+ := \sum_{a_i > 0} a_i p^i$ and $\nu_- := \sum_{a_i < 0} (-a_i) p^i$. For non-negative integers m and n satisfying $\beta - \gamma = \nu$, it holds that*

$$\max(\|\beta\|_p, \|\gamma\|_p) \geq \frac{1}{2}(\|\nu_+\|_p + \|\nu_-\|_p)$$

Proof. The assertion is a consequence of

$$\|\beta\|_p + \|\gamma\|_p \geq \|\nu_+\|_p + \|\nu_-\|_p. \tag{15}$$

Let $\beta = \sum b_i p^i$ and $\gamma = \sum c_i p^i$ where $0 \leq b_i < p$ and $0 \leq c_i < p$ be p-adic expansion of β and γ, respectively. We show the minimum value of $\|\beta\|_p + \|\gamma\|_p$ under $\beta - \gamma = \nu$ is attained with β and γ satisfying

$$b_i c_i = 0, \quad 0 \leq b_i < \frac{p}{2} \text{ and } 0 \leq c_i < \frac{p}{2} \tag{16}$$

for all $i \geq 0$. In order to prove this assertion, for given β and γ which satisfy (16) for $0 \leq i < k$ (the case $k = 0$ is allowed), we construct β' and γ' which satisfies (16) for $0 \leq i \leq k$. and $\|\beta'\|_p + \|\gamma'\|_p \leq \|\beta\|_p + \|\gamma\|_p$. Put $s_k := \min(b_k, c_k)$. In case of $b_k - s_k > \frac{p}{2}$, we put

$$\beta' := \beta + (p - b_k)p^k \quad \text{and} \quad \gamma' := \gamma + (p - b_k)p^k.$$

Then $\|\beta'\|_p \leq \|\beta\|_p + 1 - b_k$ and $\|\gamma'\|_p \leq \|\gamma\|_p + p - b_k$. (Note $b_k - s_k > 0$ implies that $c_k = s_k$ and thus $\gamma' = (\gamma - c_k p^k) + (p - (b_k - s_k))p^k$.) Therefore, $\|\beta'\|_p + \|\gamma'\|_p \leq \|\beta\|_p + \|\gamma\|_p + 1 + p - 2b_k \leq \|\beta\|_p + \|\gamma\|_p$. In case of $c_k - s_k > \frac{p}{2}$, we put

$$\beta' := \beta + (p - c_k)p^k \quad \text{and} \quad \gamma' := \gamma + (p - c_k)p^k.$$

Then, by a similar argument, we obtain $\|\beta'\|_p + \|\gamma'\|_p \leq \|\beta\|_p + \|\gamma\|_p$. Otherwise, we put

$$\beta' := \beta - s_k p^k \quad \text{and} \quad \gamma' := \gamma - s_k p^k.$$

Then $\|\beta'\|_p \leq \|\beta\|_p - s_k$ and $\|\gamma'\|_p \leq \|\gamma\|_p - s_k$. In the all three cases, the conditions (16) for β' and γ' hold for $0 \leq i \leq k$.

However, the integers β and γ satisfying (16) for all $i \geq 0$ and $\beta - \gamma = \nu$ are unique, namely $(\beta, \gamma) = (\nu_+, \nu_-)$. Therefore, (15) holds for all non-negative integers β and γ satisfying $\beta - \gamma = \nu$. □

5 Conclusion

We generalized Cheon's algorithm for divisors of $\varphi_n(p)$ to the case $n \geq 2$. In case of $n = 2$, our generalization reduces number of input data as many as half of Cheon's original algorithm. The efficiency of $n \geq 3$ is open. Until we have an efficient algorithm to test whether there is a serious security loss or not, it would be prudent to increase a side of p by, say, 20bits.

References

1. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y.G. (ed.) Public Key Cryptography - PKC 2003, Lect. Notes in Comput. Sci., vol. 2567, pp. 31-46. Springer, Berlin, Heidelberg(2002). doi: 10.1007/3-540-36288-6_3
2. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (ed.) Eurocrypt 2004, Lect. Notes in Comput. Sci., vol. 3027, pp. 56-73. Springer, Berlin, Heidelberg(2004).
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) Eurocrypt 2005, Lect. Notes in Comput. Sci., vol. 3494, pp. 440-456. Springer, Berlin, Heidelberg. doi: 10.1007/11426639_26
4. Cheon, J.H.: Security analysis of the strong Diffie-Hellman problem. In: Vaudenay, S. (ed.) Eurocrypt 2006, Lect. Notes in Comput. Sci., vol. 4004, pp. 1-11. Springer, Berlin, Heidelberg(2006). doi: 10.1007/11761679_1
5. Kozaki, S., Kutsuma, T., Matsuo, K.: Remarks on Cheon's algorithm for pairing related problems. In: Pairing-based cryptography – Pairing 2007, Lect. Notes in Comput. Sci., vol. 4575, pp. 302-316. Springer, Berlin-Heidelberg(2007). doi: 10.1007/978-3-540-73489-5_17
6. Lenstra, H.W. Jr.: Factoring integers with elliptic curves. Ann. Math. 126, 649-673 (1987).
7. Nechaev, V.I.: On the complexity of a deterministic algorithm for a discrete logarithm. Mat. Zametki 55, 91-101 (1994), translation in Math. Notes 55(1994) 165-172.
8. Shanks, D.: Class number, a theory of factorization, and genera. In: 1969 Number Theory Institute, Proc. Symp. Pure. Math., vol. 20, pp. 415-440. AMS, Providence, R.I.(1971).
9. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Advances in cryptology - EUROCRYPT'97, Lect. Notes in Comput. Sci., vol. 1233, pp. 256-266. Springer, Berlin(1997). doi: 10.1007/3-540-69053-0_18