IEICE TRANS. FUNDAMENTALS, VOL.E87–A, NO.1 JANUARY 2004

PAPER *Special Section on Cryptography and Information Security*

Analysis of Baby-Step Giant-Step Algorithms for Non-uniform Distributions

Koh-ichi NAGAO[†a)], **Shigenori UCHIYAMA**[††], **Naoki KANAYAMA**[†††], *and* **Kazuto MATSUO**[††††], ***Members***

SUMMARY The baby-step giant-step algorithm, BSGS for short, was proposed by Shanks in order to compute the class number of an imaginary quadratic field. This algorithm is at present known as a very useful tool for computing with respect to finite groups such as the discrete logarithms and counting the number of the elements. Especially, the BSGS is normally made use of counting the rational points on the Jacobian of a hyperelliptic curve over a finite field. Indeed, research on the practical improvement of the BSGS has recently received a lot of attention from a cryptographic viewpoint. In this paper, we explicitly analyze the modified BSGS, which is for non-uniform distributions of the group order, proposed by Blackburn and Teske. More precisely, we refine the Blackburn-Teske algorithm, and also propose a criterion for the decision of the effectiveness of their algorithm; namely, our proposed criterion explicitly shows that what distribution is needed in order that their proposed algorithm is faster than the original BSGS. That is, we for the first time present a necessary and sufficient condition under which the modified BSGS is effective.

key words: *baby-step giant-step algorithm, finite group*

1. Introduction

The baby-step giant-step algorithm, BSGS for short, was proposed by D. Shanks [6] in order to compute the class number of an imaginary quadratic field. The BSGS is at present known as a useful and efficient tool for counting the number of elements of an arbitrary finite group; e.g. Mestre employed the BSGS for counting the number of points on an elliptic curve defined over a finite field (see [3]). Especially, the BSGS is often made use of counting the rational points on the Jacobian of a curve over a finite field for high genus curve based cryptography. Here we note that the most notable recent improvement of BSGS can be found in [5], which gives an essential improvement of [4].

Since Shanks' proposal of the BSGS, some improvements and modifications have been investigated by a lot of researchers. In [2], Buchmann et al. proposed an improvement of the original method based on the following idea: First, divide the search interval into a lot of "short" intervals, then put a suitable "baby-step depth," which is the number of baby-steps for an individual short interval. Buchmann et al. showed its effectiveness by implementation on computing the class number of imaginary quadratic number fields. Besides, their method can be applied to three representative problems with respect to finite groups: computing the discrete logarithm, computing the number of elements, and determining structure of a group. Furthermore, Terr [8] improved their method to apply to computing the class number of imaginary quadratic number fields. Note that the improvement of Buchmann et al. and Terr assumes that the distribution of the number of the group elements is uniform.

Recently, Blackburn and Teske [1] have proposed an improvement of [2] in the case that the distribution of the number of the group elements is known and is not always uniform. Although, [1] only concerned with the discrete logarithm problem, it is easy to see that the algorithms in [1] can be used for counting the number of elements of a group as well as computing the discrete logarithms. In [1], they also considered the way of optimization for the baby-step depth of an individual short interval by using their proposed "baby-step function." Namely, if the distribution of the answers of the target problems is given, we might be able to take the optimal baby-step depth for an individual short interval. However, Blackburn and Teske only gave a practical consideration and mentioned "We use the practical algorithm with short cut; by doing this, we risk some discrepancy with our theoretically optimal function, but this is the algorithm that would be used in practice after all. (page 165 in [1])," so, they would not explicitly describe the way of optimization.

In this paper, we present a refinement of the modified BSGS proposed by Blackburn and Teske. We also propose a criterion for the decision of the effectiveness of the Blackburn-Teske algorithm. Namely, our proposed criterion explicitly shows that what distribution is needed in order that their proposed algorithm is faster than the original BSGS. That is, we for the first time present a necessary and sufficient condition

Manuscript received March 24, 2003.
Manuscript revised June 30, 2003.
Final manuscript received August 7, 2003.

[†]The author is with the Faculty of Engineering, Kanto Gakuin University, Yokohama-shi, 236-8501 Japan.
[††]The author is with NTT Information Sharing Platform Laboratories, Yokosuka-shi, 239-0847 Japan.
[†††]The author is with the Department of Information and Communication Engineering, The University of Electro-Communications, Chofu-shi, 182-8585 Japan.
[††††]The author is with the Research and Development Initiative, Chuo University, Tokyo, 112-8851 Japan.
a) E-mail: nagao@kanto-gakuin.ac.jp

under which the modified BSGS is effective. When it comes to the application of the modified BSGS and our proposed criterion to cryptography, we show that the modified BSGS would not be more effective than what is expected in [1] in the case of computation of the Jacobian order of a genus 2 hyperelliptic curve over a finite field by using experimental results. However, this does not deny the existence of positive contribution of the modified BSGS to cryptography. So one hopes the modified BSGS can be useful to some expected cryptographic primitives in time to come.

This paper is organized as follows. First, we will review the original BSGS in Sect. 2. Second, we will analyze the Blackburn-Teske modified BSGS in Sect. 3. Finally we conclude this paper.

Throughout this paper, we assume that the distribution of the group order is given, and the efficiency of algorithms on a finite group is estimated by the average of the number of additions on a group. So, we assume that the efficiency of the inverse operations on a group can be negligible.

2. The Baby-Step Giant-Step Algorithm

In this section, we will review the original BSGS algorithm, and give some notations and definitions.

We assume that the following situation: Computing the order of a finite group G; we are given the distribution of the order of G's, and G is from a set of groups $\mathcal{G} = \{G\}$ satisfying $|\#G - c| < w$ where $c, w \in \mathbb{Z}$ are some constants. From here on, w.l.o.g., we may also assume that each G is cyclic.

We also assume that the cost of one baby step is the same as that of one giant step. We will discuss the case that the cost of one baby step is cheaper than that of giant step in Appendix B.

Algorithm 1 Baby-Step Giant-Step

Input: A finite cyclic group G.
Output: The order of G.
1: Choose a random element $P \in G$.
2: Compute $BS = [\sqrt{w}]$.
3: Compute $(c+j)P$ $(j = 0, 1, 2, \ldots, BS-1)$, and store them.
4: Compute $BS \cdot P$.
5: **for** $i = 0, \ldots, \lceil w/BS \rceil$ **do**
6: Compute $Q = i \cdot (BS \cdot P)$.
7: **if** $^{\exists} j$ such that $Q = \pm(c+j)P$ **then**
8: **return** $c + j \mp (i \cdot BS)$ as (a candidate for) $\#G$.
9: **end if**
10: **end for**

Here we note the group operation on G is additively written. Put M be the average of $|\#G - c|$'s, then we can optimize the number of additions on G that the BSGS requires, say T, for the original BSGS to be $2\sqrt{M}$, where we take BS, the baby-step depth, i.e. the number of the baby-steps, as approximately $\sqrt{M}$. Actually, let $f(t)$ be the probability function associated with the distribution of $\{|\#G - c|\}$, i.e.,

$$f(t) = \mathrm{Prob}[\, G \in \mathcal{G} \mid l(G) = t \,],$$

where $l(G) = |\#G - c|$, then we have $M = \sum_{t=0}^{w-1} t f(t)$, and the number of additions on G in the giant-step is expected to be on the order of $\frac{x}{BS}$ when $x = |\#G - c|$, where BS denotes the baby-step depth. We also have $T = BS + \sum_{t=0}^{w-1} \frac{t}{BS} f(t) = BS + \frac{M}{BS} \geq 2\sqrt{M}$, where the equality holds if and only if $BS = \sqrt{M}$.

3. BSGS for Non-uniform Distributions

In this section, we discuss the modified BSGS proposed by Blackburn-Teske [1].

The notations are as in the previous section. First of all, we define the functions $p(x), M(x)$ as $p(x) = \sum_{t=0}^{x-1} f(t)$, $M(x) = \sum_{t=0}^{x-1} t f(t)$, $M = \sum_{t=0}^{w-1} t f(t)$.

Put $I = [0, w)$. Now, we take a division, Δ, of I:

$$\Delta : 0 = w_0 < w_1 < w_2 < \cdots < w_n = w, \quad (w_i \in \mathbb{Z})$$

where n is a positive integer. Put $|\Delta| = n$. Furthermore, put $I_{\Delta,i} = [w_{i-1}, w_i)$, and

$$h_{\Delta,i} = w_i - w_{i-1}, \quad P_{\Delta,i} = \sum_{t=w_{i-1}}^{w_i - 1} f(t).$$

Note that $P_{\Delta,i}$ denotes the probability that $|\#G - c|$ lies in the i-th cell $I_{\Delta,i}$. We call that Δ is a partition, if $P_{\Delta,i} > 0$ for each i. Further, we call each $I_{\Delta,i}$ the "i-th cell" of the division (resp. partition) Δ of I.

For simplicity, we also write the division (resp. partition) Δ as follows:

$$I = I_{\Delta,1} + \cdots + I_{\Delta,n} \quad \text{or} \quad I = \sum_{i=1}^{n} I_{\Delta,i}.$$

Here we call $h_{\Delta,i}$ the "i-th depth" of the division (resp. partition) Δ of I. Note that Blackburn and Teske [1] only considered the case that all $h_{\Delta,i}$ are the same.

Now, let $BS_{\Delta,i}$ $(i = 1, \cdots, n)$ be positive integers satisfying the following:

$$0 < BS_{\Delta,1} \leq BS_{\Delta,2} \leq \ldots \leq BS_{\Delta,n}.$$

We will soon give the modified BSGS algorithm, for a partition Δ, using $BS_{\Delta,i}$ as the baby-step depth for the cell $I_{\Delta,i}$. In that case, we call the $BS_{\Delta,i}$ the baby-step depth of the i-th cell of partition Δ. Note that $BS_{\Delta,i}$ is called baby-step function in [1].

Also, we usually put $BS_i = BS_{\Delta,i}$, $I_i = I_{\Delta,i}$, $P_i = P_{\Delta,i}$ and $BS_0 = 0$ below. Besides, for simplicity, we sometimes set $h_{\Delta,i} = h_i$ unless we are confused.

Algorithm 2 Modified baby-step giant-step

Input: A finite cyclic group G.
Output: The order of G.

```
Choose a random element P ∈ G.
for k = 1, ..., n do
    Compute (c + j)P (j = BS_{k-1}, BS_{k-1} + 1, ..., BS_k − 1) and store them.
    Compute BS_k · P.
    for i = 0, ..., ⌈h_k / BS_k⌉ do
        Compute Q = i · (BS_k · P) + w_{k-1} P.
        if ∃j such that 0 ≤ j ≤ BS_k − 1 and Q = ±(c + j)P then
            return c + j ∓ (i · BS_k + w_{k-1}) as ( a candidate for ) #G.
        end if
    end for
end for
```

The total running time of this algorithm can be easily estimated as in [1] (page 156):

$$\sum_{j=1}^{n}\left\{P_j\cdot(BS_j+A_j)+\left\lceil\frac{M_j}{BS_j}\right\rceil\right\}$$

where

$$A_j=\begin{cases}\sum_{k=1}^{j-1}\lceil\frac{h_k}{BS_k}\rceil & j=1,\ldots,n-1\\ 0 & j=n\end{cases}\quad\text{and}$$

$$M_j=\sum_{t=w_{j-1}}^{w_j-1}(t-w_{j-1})f(t)=\sum_{t=w_{j-1}}^{w_j-1}tf(t)-w_{j-1}P_j.$$

Asymptotically, the running time is expected to be

$$T_\Delta=\sum_{k=1}^{n}\left(P_k\cdot BS_k+\frac{B_k}{BS_k}\right),$$

where

$$B_k=\begin{cases}M_k+h_k\sum_{j=k+1}^{n}P_j & k=1,\ldots,n-1\\ M_k & k=n\end{cases}.$$

Note that $\frac{M_k}{BS_k}$ is the average of the running time of the giant-step for $I_{\Delta,k}$ in the case that $|\#G-c|$ lies in the k-th interval.

We will study the property of the function T_Δ in order to make a criterion for the decision of the effectiveness of this algorithm, which is the main purpose of this paper. More precisely, we will study the condition that there is a certain partition Δ and real numbers $0<s_1<\cdots<s_n$ such that $T_\Delta(s_1,\cdots,s_n)\le 2\sqrt{M}$. Here, we know that the running time of original BSGS algorithm is roughly $2\sqrt{M}$.

Put

$$t_k(s)=P_ks+\frac{B_k}{s},\quad m_k=\sqrt{\frac{B_k}{P_k}},$$

where s is an arbitrary positive real number.

Let T_Δ be a function defined by the following:

$$T_\Delta(s_1,s_2,\ldots,s_n)=\sum_{i=1}^{n}t_i(s_i),$$

where $s=(s_1,s_2,\ldots,s_n)$ from $\mathbb{R}^n_{>0}$. Besides, we can describe the functions P_k,B_k by using the probability function $f(t)$ as follows.

$$P_k=\sum_{t=w_{k-1}}^{w_k-1}f(t),$$

$$B_k=h_k\sum_{t=w_k}^{w-1}f(t)+\sum_{t=w_{k-1}}^{w_k-1}(t-w_{k-1})f(t).$$

These numbers P_k,B_k only depend on the interval I_k. So, when an interval $J=[\alpha,\beta)$ $(0\le\alpha<\beta\le w,\quad \alpha,\beta\in\mathbb{Z})$ is given, then we can define P_J,B_J by the following:

$$P_J=\sum_{t=\alpha}^{\beta-1}f(t),$$

$$B_J=(\beta-\alpha)\sum_{t=\beta}^{w-1}f(t)+\sum_{t=\alpha}^{\beta-1}(t-\alpha)f(t).$$

With these notations, we note that $P_k=P_{I_k},B_k=B_{I_k}$. Similarly, we define $t_J(s)=P_Js+\frac{B_J}{s}$. Also, we define $m_J=\sqrt{\frac{B_J}{P_J}}$, if $P_J>0$.

We will study the property of the function T_Δ.

Lemma 3.1.

$$T_\Delta(m_1,\ldots,m_n)=2\sum_{i=1}^{n}P_i\cdot m_i.$$

Proof. We can obtain above equation immediately from the definition. □

Lemma 3.2. T_Δ has the unique minimum at $(m_1,\ldots,m_n)$. Namely, for any $(s_1,\ldots,s_n)\in\mathbb{R}^n_{>0}$,

$$T_\Delta(s_1,\ldots,s_n)\ge T_\Delta(m_1,\ldots,m_n).$$

The equality holds if and only if $s_1=m_1,\ldots,s_n=m_n$.

Proof. $T_\Delta(s_1,\ldots,s_n)$ is of the form $\sum t_i(s_i)$ and each $t_i(s_i)$ has the unique minimum at $s_i=m_i$. □

Lemma 3.3.

$$\sum_{i=1}^{n}P_i=1,\qquad\sum_{i=1}^{n}B_i=M.$$

Proof. It is easy to see that $\sum_{i=1}^{n}P_i=\sum_{t=0}^{w-1}f(t)=1$. Since $M_k=\sum_{t=w_{k-1}}^{w_k-1}tf(t)-w_{k-1}P_k=\sum_{t=w_{k-1}}^{w_k-1}tf(t)-P_k\sum_{j=1}^{k-1}h_j$, we have $\sum_{k=1}^{n}M_k=\sum_{t=0}^{w-1}tf(t)-\sum_{k=1}^{n}\sum_{j=1}^{k-1}P_kh_j=M-\sum_{j=1}^{n-1}\sum_{k=j+1}^{n}P_kh_j$ and we easily obtain the formula of $\sum B_i$. □

For a partition Δ, we will consider the interval $I_{new} = \cup_{i=k_1}^{k_2} I_{\Delta,i}$.

Lemma 3.4.

$$P_{I_{new}} = \sum_{i=k_1}^{k_2} P_i, \quad B_{I_{new}} = \sum_{i=k_1}^{k_2} B_i,$$
$$t_{I_{new}}(s) = \sum_{i=k_1}^{k_2} t_i(s)$$

Proof. Here we only show the second one; $B_{[\alpha,\beta]} + B_{[\beta,\gamma]} = B_{[\alpha,\gamma]}$, because the others are easily obtained from the representation of P_k and $t_k(s)$ similarly. Since $B_{[\alpha,\beta]} = (\beta-\alpha)\sum_{t=\beta}^{w-1} f(t) + \sum_{t=\alpha}^{\beta-1}(t-\alpha)f(t) = (\beta-\alpha)\sum_{t=\gamma}^{w-1} f(t)+(\beta-\alpha)\sum_{t-\beta}^{\gamma-1} f(t)+\sum_{t=\alpha}^{\beta-1}(t-\alpha)f(t)$ and $B_{[\beta,\gamma]} = (\gamma-\beta)\sum_{t=\gamma}^{w-1} f(t) + \sum_{t=\beta}^{\gamma-1}(t-\beta)f(t) = (\gamma-\beta)\sum_{t=\gamma}^{w-1} f(t)+\sum_{t=\beta}^{\gamma-1}(t-\alpha)f(t)-(\beta-\alpha)\sum_{t=\beta}^{\gamma-1} f(t)$, we can easily obtain the above equation. □

Since Δ is a partition, we see $P_i > 0$ for each i and $P_{I_{new}} > 0$. So, $m_{I_{new}}$ is well defined.

Lemma 3.5.

$$\min\{m_{k_1}, \ldots, m_{k_2}\} \le m_{I_{new}} \le \max\{m_{k_1}, \ldots, m_{k_2}\}.$$

The equality holds if and only if $m_{k_1} = \cdots = m_{k_2}$.

Proof. It is easy to see that $m_{I_{new}}^2 = \frac{\sum B_i}{\sum P_i} = \frac{\sum P_i \cdot m_i^2}{\sum P_i} \le \max\{m_{k_1}^2, \ldots, m_{k_2}^2\}$ and also the equality holds iff $m_{k_1} = \cdots = m_{k_2}$. Similarly, we have $\min\{m_{k_1}, \ldots, m_{k_2}\} \le m_{I_{new}}$. □

Lemma 3.6.

$$T_\Delta(\sqrt{M}, \ldots, \sqrt{M}) = 2\sqrt{M}, \text{and}$$
$$T_\Delta(m_1, \ldots, m_n) \le 2\sqrt{M},$$

where the equality of the latter formula holds if and only if $m_1 = \ldots = m_n = \sqrt{M}$.

Proof. From Lemma 3.3, we have

$$T_\Delta(\sqrt{M}, \ldots, \sqrt{M}) = \sum P_i \sqrt{M} + \frac{\sum B_i}{\sqrt{M}} = 2\sqrt{M}.$$

So, we have the former equation. On the other hand, from Lemma 3.2, we have

$$2\sqrt{M} = T_\Delta(\sqrt{M}, \ldots, \sqrt{M}) \ge T_\Delta(m_1, \ldots, m_n),$$

and the equality of the latter formula holds if and only if $m_1 = \ldots = m_n = \sqrt{M}$. □

Now, for a partition Δ of I, we define the "effectiveness" of the partition Δ as follows.

Definition 3.7. For any partition Δ of I, we say Δ is effective if and only if $m_1 < \cdots < m_n$.

We can decide whether the modified BSGS algorithm is faster than the original one or not by using this property of m_i's.

If a partition Δ is effective, taking the baby-step depth $BS_i = m_i$, if we ignore the condition BS_i's being integers, the running time of the algorithm $T_\Delta(BS_1, \cdots, BS_n)$ is shorter than $2\sqrt{M}$ (c.f. Lemma 3.6). Since the running time of the original BSGS algorithm is expected to be $2\sqrt{M}$, the existence of an effective partition is sufficient condition that the running time of this algorithm is shorter than that of the original one. Furthermore, we will show that these conditions are equivalent in Theorem 3.11.

Lemma 3.8. Let Δ be an effective partition of I. Let Δ^* be a partition of I of the form

$$I_{\Delta^*,i} = \begin{cases} I_{\Delta,i} & i < k_1 \\ \cup_{j=k_1}^{k_2} I_{\Delta,j} & i = k_1 \\ I_{\Delta,i+k_2-k_1} & k_1 + 1 < i \le n - k_2 + k_1 \end{cases}.$$

Then the partition Δ^* is also effective.

Proof. This lemma is followed by the relation $m_{\Delta,k_1} < m_{I_{\Delta^*,k_1}} < m_{\Delta,k_2+1}$, which is due to Lemma 3.5. □

Lemma 3.9. Let Δ be a partition of I consist of 2 cells, i.e. $I = I_1 + I_2$. Then we have

$$m_1 < \sqrt{M} \text{ if and only if } m_2 > \sqrt{M},$$

and

$$m_1 > \sqrt{M} \text{ if and only if } m_2 < \sqrt{M}.$$

Proof. By Lemma 3.3, $P_2 = 1 - P_1$, $B_2 = M - B_1$ and $m_2 = \sqrt{\frac{M-B_1}{1-P_1}}$. Then, we see easily that the condition $m_1 = \sqrt{\frac{B_1}{P_1}} < \sqrt{M}$ is equivalent to the conditions $M - B_1 > M - P_1 M$ and $m_2 = \sqrt{\frac{M-B_1}{1-P_1}} > \sqrt{M}$. So, we have the former relation. Similarly we can obtain the other relation. □

Let $x_{min} = \min\{t \in \mathbb{Z} | f(t) > 0\} + 1$ and $x_{max} = \max\{t \in \mathbb{Z} | f(t) > 0\}$.

The partition Δ of I, consist of 2 cells is of the form $I = [0, x) + [x, w)$ for some integer x with $x_{min} \le x \le x_{max}$. Besides, for any integer x with $x_{min} \le x \le x_{max}$, the division Δ of $I = [0, x) + [x, w)$ is a partition. Remark that $m_1 = m_{[0,x)}$ is of the form

$$\sqrt{\frac{(1-p(x))x + M(x)}{p(x)}} \qquad (x_{min} \le x \le x_{max}).$$

Example 1. We consider the case $w = 10000$, $I = [0, 10000)$, $I_1 = [0, 1000)$, $I_2 = [1000, 10000)$.

$$f(x) = \begin{cases} 11/20000 & x \in I_1 \cap \mathbb{Z} \\ 1/20000 & x \in I_2 \cap \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}.$$

Then we have $P_1 = 11/20, P_2 = 9/20$, and $2\sqrt{M} = 2\sqrt{\sum_{t=0}^{9999} tf(t)} = 104.87$.
Taking the partition Δ of the form $I = I_1 + I_2$, we have $m_1 = 36.30$ and $m_2 = 67.08$. So, this is an effective partition. In fact, $T_\Delta = 100.3$ and approximately 4 percent speed up is done.

The following theorem gives a characterization that, for a given probability function, there exists an effective partition or not.

Theorem 3.10. The following three conditions are equivalent:
(a) There exists an effective partition, Δ, satisfying $|\Delta| \geq 2$.
(b) There exists a positive integer x such that $m_{[0,x)} < \sqrt{M}$ where $x_{min} \leq x \leq x_{max}$.
(c) There exists a positive integer x such that $m_{[x,w)} > \sqrt{M}$ where $x_{min} \leq x \leq x_{max}$.

Proof. For any integer x with $x_{min} \leq x \leq x_{max}$, the division Δ of $I = [0, x) + [x, w)$ is a partition. So, obviously (b) and (c) are equivalent due to Lemma 3.9. So, we only need to show (a) and (b) are equivalent.

(a) $\rightarrow$ (b): By Lemma 3.8, there is an effective partition Δ^* of $I = I_1 + I_2$. Then, there is some integer x with $x_{min} \leq x \leq x_{max}$ such that $I_1 = [0, x)$. Also, due to Lemma 3.1 and 3.6, $2\sqrt{M} \geq T_{\Delta^*}(m_1, m_2) = 2P_1 m_1 + 2P_2 m_2 > 2m_1$, so we have (b).

(b) $\rightarrow$ (a): Put $I_1 = [0, x), I_2 = [x, w)$. Since $x_{min} \leq x \leq x_{max}$, the division Δ of $I = I_1 + I_2$ is a partition. From Lemma 3.9, it is an effective partition. □

Example 2. As an application of this theorem, we could apply this to compute the order of the Jacobian of genus 2 hyperelliptic curve. By some experiments, see Appendix, we have $m_{[0,x)} = \sqrt{\frac{(1-p(x))x+M(x)}{p(x)}} > \sqrt{M}$. So, we may conclude that there exist no effective partitions on the distribution of the orders of the Jacobian of genus 2 hyperelliptic curves defined over a fixed finite field.

The following theorem asserts that the existence of an effective partition is necessary condition for that the running time of the modified BSGS is shorter than that of original one. Note that the sufficiency is directly obtained from Lemma 3.6 and Definition 3.7. So, these conditions are equivalent. Further, from Theorem 3.10, this condition can be reduced to the computation of $m_{[0,x)} = \sqrt{\frac{(1-p(x))x+M(x)}{p(x)}}$ and we can decide the criterion that the modified BSGS algorithm is faster than original one.

Theorem 3.11. Assume that, for a given probability function, there exist no partitions satisfying the condition (a) of Theorem 3.10. Then, for any partition Δ, where $|\Delta| = n$ and any positive real numbers $S_1, \ldots, S_n$ such that $0 < S_1 \leq S_2 \leq \cdots \leq S_n$, we have

$$T_\Delta(S_1, S_2, \ldots, S_n) \geq 2\sqrt{M}.$$

Proof. Suppose that there exists a partition Δ, $|\Delta| = n$, and some positive real numbers $S_1, \ldots, S_n$ such that $0 < S_1 \leq S_2 \leq \cdots \leq S_n$ satisfying

$$T_\Delta(S_1, S_2, \ldots, S_n) < 2\sqrt{M}. \quad (1)$$

Furthermore, we may assume that n is the smallest one among the partitions satisfying above assumption. Then, n must be greater than or equal to 2. Actually, if $n = 1$, then we have $T_\Delta(s) = s + \frac{M}{s} \geq 2\sqrt{M}$ this contradicts the assumption.

Also we need the following three claims in order to prove this theorem.

Claim A. The function $T_\Delta(s_1, \ldots, s_n)$ has the minimum value on the domain $D = \{(s_1, \ldots, s_n) \in \mathbb{R}^n_{>0} \mid s_1 \leq \ldots \leq s_n\}$.

Proof. Let $\delta = \max(\frac{2\sqrt{M}}{P_n}, \frac{2\sqrt{M}}{B_1}, \sqrt{M}, \frac{1}{\sqrt{M}})$ and $D_\delta = \{(s_1, s_2, \cdots, s_n) \in \mathbb{R}^n_{>0} \mid \frac{1}{\delta} \leq s_1 \leq s_2 \leq \cdots \leq s_n \leq \delta\}$.

For any $Q = (s_1, \cdots, s_n) \in D \backslash D_\delta$, from the definition of D_δ, we see $0 < s_1 < \frac{1}{\delta}$ or $s_n > \delta$. When $0 < s_1 < \frac{1}{\delta}$, we see $\frac{1}{s_1} > \delta$ and $\frac{B_1}{s_1} > B_1 \cdot \delta \geq 2\sqrt{M}$ and $T_\Delta(Q) > t_1(s_1) > \frac{B_1}{s_1} > 2\sqrt{M}$.
When $s_n > \delta$, we see $P_n \cdot s_n > P_n \cdot \delta \geq 2\sqrt{M}$ and $T_\Delta(Q) > t_n(s_n) > P_n s_n > 2\sqrt{M}$.
So, we obtain $T_\Delta(Q) > 2\sqrt{M}$ for any $Q \in D \backslash D_\delta$. On the other hand, D_δ is a compact bounded domain in $\mathbb{R}^n_{>0}$. So T_Δ has the minimum value at P_0 in the domain D_δ, i.e.,

$$T_\Delta(P_0) \leq T_\Delta(P) \text{ for any} P \in D_\delta.$$

Put $P_1 = (\sqrt{M}, \cdots, \sqrt{M})$. We easily see $P_1 \in D_\delta$ and $T_\Delta(P_1) = 2\sqrt{M}$ from Lemma 3.6. So, we have $T_\Delta(P_0) \leq T_\Delta(P_1) = 2\sqrt{M} < T_\Delta(Q)$ where Q is any element of $D \backslash D_\delta$. So, it is obvious that T_Δ has the minimum value on the domain D. □

Suppose that T_Δ has the minimum value at $(S'_1, \ldots, S'_n) \in D$. Note that $0 < S'_1 \leq S'_2 \leq \cdots \leq S'_n$ and $T_\Delta(S'_1, S'_2, \ldots, S'_n) < 2\sqrt{M}$.

Claim B. $S'_1 < S'_2 < \ldots < S'_n$.

Proof. For some i, we assume $S'_i = S'_{i+1}$. Taking a new partition Δ^* of the form

$$I_{\Delta^*,j} = \begin{cases} I_{\Delta,j} & j < i \\ I_{\Delta,i} \cup I_{\Delta,i+1} & j = i \\ I_{\Delta,i+1} & i < j \leq n-1 \end{cases}.$$

Note that $|\Delta^*| = n - 1$. Then, by Lemma 3.4, $t_{I_{\Delta^*,i}}(s) = t_{\Delta,i}(s) + t_{\Delta,i+1}(s)$, and we have $T_{\Delta^*}(S'_1, \ldots, S'_i, S'_{i+2}, \ldots, S'_n) = T_\Delta(S'_1, \ldots, S'_n) < 2\sqrt{M}$. This contradicts the minimum of n. □

Claim C. $S'_1 \geq \sqrt{M}$.

Proof. Suppose that $S'_1 < \sqrt{M}$. Put $I_{\Delta,1} = [0, x)$, then we have $m_1 = m_{[0,x)} \geq \sqrt{M}$, since the condition (b) of Theorem 3.10, which is equivalent to the condition (a), does not hold. Therefore, the function $t_1(s) = P_1(s + \frac{m_1^2}{s})$ is monotone decreasing on $(0, \sqrt{M}]$. Let $e = \min\{S'_2 - S'_1, \sqrt{M} - S'_1\}$ ($e > 0$ due to Claim B), then we have $t_1(S'_1 + e) < t_1(S'_1)$. As a result, we obtain

$$\begin{aligned} &T_\Delta(S'_1 + e, S'_2, \ldots, S'_n) - T_\Delta(S'_1, \ldots, S'_n) \\ &\quad = t_1(S'_1 + e) - t_1(S'_1) < 0. \end{aligned}$$

This contradicts the minimum of the value $T_\Delta(S'_1, \ldots, S'_n)$. □

Claim D. $S'_n \leq \sqrt{M}$.

Proof. Suppose that $S'_n > \sqrt{M}$. Put $I_{\Delta,n} = [x, w)$, then we have $m_n = m_{[x,w)} \leq \sqrt{M}$, since the condition (c) of Theorem 3.10, which is equivalent to the condition (a), does not holds. Therefore, the function $t_n(s) = P_n(s + \frac{m_n^2}{s})$ is monotone increasing on $[\sqrt{M}, \infty)$. Let $e = \min\{S'_n - S'_{n-1}, S'_n - \sqrt{M}\}$ ($e > 0$ due to Claim B), then we have $t_n(S'_n - e) < t_n(S'_n)$. As a result, we obtain

$$\begin{aligned} &T_\Delta(S'_1, S'_2, \ldots, S'_n - e) - T_\Delta(S'_1, \ldots, S'_n) \\ &\quad = t_n(S'_n - e) - t_n(S'_n) < 0. \end{aligned}$$

This contradicts the minimum of the value $T_\Delta(S'_1, \ldots, S'_n)$. □

Finally, above claims B, C and D are obviously contradict each other. So, there are no positive real numbers satisfying $0 < S_1 \leq S_2 \leq \cdots \leq S_n$, and $T_\Delta(S_1, S_2, \ldots, S_n) \geq 2\sqrt{M}$. This completes the proof of theorem. □

From this theorem, we can conclude that an effective partition must be needed for that the modified BSGS algorithm is faster than the original one.

4. Conclusion

In this paper, we explicitly analyzed the modified BSGS proposed by Blackburn and Teske. More precisely, we refined the Blackburn-Teske algorithm, and also proposed a criterion for the decision of the effectiveness of their algorithm. Our proposed criterion explicitly shows that what distribution is needed so that their proposed algorithm is faster than the original BSGS. That is, we for the first time present a necessary and sufficient condition under which the modified BSGS is effective.

Acknowledgment

The authors would like to thank the editor and the anonymous referees for their useful comments. This research was partially supported by the Telecommunications Advancement Organization of Japan (TAO).

References

[1] S.R. Blackburn and E. Teske, "Baby-step giant-step algorithms for non-uniform distributions," Proc. ANTS-IV, LNCS 1838, pp.153–168, Springer-Verlag, 2000.
[2] J. Buchmann, M.J. Jacobson, Jr., and E. Teske, "On some computational problems in finite abelian groups," Math. Comput., vol.66, pp.1663–1687, 1997.
[3] H. Cohen, A Course in Computational Algebraic Number Theory, GTM 138, Springer-Verlag, 1993.
[4] P. Gaudry and R. Harley, "Counting points on hyperelliptic curves over finite fields," Proc. ANTS-IV, LNCS 1838, pp.313–332, Springer-Verlag, 2000.
[5] K. Matsuo, J. Chao, and S. Tsujii, "An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields," Proc. ANTS-V, LNCS 2369, pp.461–474, Springer-Verlag, 2002.
[6] D. Shanks, "Class number, a theory of factorization, and genera," Proc. Symp. Math. Soc. 20 (1969 Institute on Number Theory), pp.415–440, American Math. Society, Providence, 1971.
[7] E. Teske, "Square-root algorithms for the discrete logarithm problem (a survey)," in Public-Key Cryptography and Computational Number Theory, pp.283–301, Walter de Gruyter, Berlin/New York, 2001.
[8] D.C. Terr, "A modification of Shanks' baby-step giant-step algorithm," Math. Comput., vol.69, pp.767–773, 2000.
[9] E. Teske and A. Stein, "Optimized baby step-giant step methods: An applications to hyperelliptic function fields," Research Report CORR 2001-62, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, 2001.

Appendix A: Distribution of the Orders of the Jacobians of Genus 2 Hyperelliptic Curves

For a fixed prime $q = 1048573$ (20 bits) (resp. 16777213 (24 bits)), we randomly generated 10000 hyperelliptic curves over $\mathbb{F}_q$ of the form $Y^2 = F(X)$ ($\deg F(X) = 5$) and computed the orders of their Jacobian groups. Note that $c = q^2 + 6q + 1$ and $w = \lceil 4\sqrt{q^3} \rceil$.

Let $\phi(x) = \sqrt{\frac{(1-p(x))x + M(x)}{p(x)}}$ as in Example 2.

Tables A·1, A·2 show the values x/w, $\phi(x)$ and $\phi(x)/\sqrt{M}$ for each q.

Appendix B: When Baby Steps are Cheaper than Giant Steps

In this paper, we consider the running time of the original BSGS, $T(BS)$ (resp. modified BSGS, $T_\Delta(BS_1, \cdots)$)

Table A· 1 The ratio $\phi(x)/\sqrt{M}$ for a 20 bits prime $q = 1048573$.

x/w	$p(x)$	$\phi(x)$	$\phi(x)/\sqrt{M}$
0.10	0.3202	33780.96669	1.142820168
0.20	0.5798	32303.94212	1.092851987
0.30	0.7675	31186.37522	1.055044366
0.40	0.8821	30522.71627	1.032592586
0.50	0.9543	29953.83883	1.013347292
0.60	0.9848	29707.81703	1.005024302
0.70	0.9966	29596.65216	1.001263561
0.80	0.9997	29562.35649	1.000103328
0.85	0.9998	29561.96792	1.000090183
0.90	1	29559.30218	1
1.00	1	29559.30218	1

Table A· 2 The ratio $\phi(x)/\sqrt{M}$ for a 24 bits prime $q = 16777213$.

x/w	$p(x)$	$\phi(x)$	$\phi(x)/\sqrt{M}$
0.10	0.324	267917.1503	1.134461128
0.20	0.5822	257007.4622	1.08826544
0.30	0.7676	248807.4965	1.053543727
0.40	0.8851	243111.581	1.029425096
0.50	0.9506	239591.8312	1.014521163
0.60	0.9845	237349.4101	1.005025916
0.70	0.9953	236577.629	1.001757907
0.80	0.9994	236219.3861	1.000240973
0.84	0.9998	236183.769	1.000090157
0.90	1	236162.4773	1
1.00	1	236162.4773	1

under the assumption that the cost of one baby step is the same as that of one giant step. [9] treats the case that the cost of one baby step is cheaper than that of giant step in order to apply BSGS for the order counting of the Jacobian of hyperelliptic curves. Assume

$$n = \frac{\text{running time of one giant step}}{\text{running time of one baby step}},$$

and we will consider the running time of the original BSGS, $T^n(BS)$ (resp. modified BSGS, $T^n_\Delta(BS_1, \cdots)$) , where the total running time is estimated by the sum of the number of baby steps and n times the number of giant step. Then, from the proof of [9, Proposition 2.5], we have

$$\begin{aligned} T^n(BS) &= BS + \frac{M}{BS} \cdot n \\ &= \sqrt{n} \cdot \left(\frac{BS}{\sqrt{n}} + \frac{M}{\frac{BS}{\sqrt{n}}} \right) = \sqrt{n} \cdot T\left(\frac{BS}{\sqrt{n}} \right) \end{aligned}$$

for original BSGS, and the estimation of the total running time $T^n(BS)$ is essentially reduced to that of $T(BS)$. Similarly, for modified BSGS, we easily have

$$T^n_\Delta(BS_1, \cdots) = \sqrt{n} \cdot T_\Delta\left(\frac{BS_1}{\sqrt{n}}, \cdots \right),$$

and the estimation of the total running time $T^n_\Delta(BS_1, \cdots)$ is essentially reduced to that of $T_\Delta(BS_1, \cdots)$, which is the main subject of this paper.

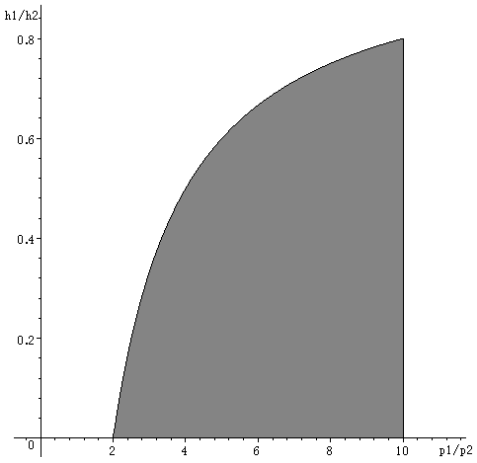


Fig. A· 1 The domain satisfying in Eq. (A· 1).

So, the statements in this paper can be easily extended to the case that the cost of one baby step is cheaper than that of giant step.

Appendix C: The Split Uniform Distribution

Suppose that the interval $I = [0, h_1 + h_2)$ is divided into two parts $I_1 = [0, h_1)$ and $I_2 = [h_1, h_1 + h_2)$. We consider "the split uniform distribution" such that

$$f(t) = \begin{cases} p_1 & t \in I_1 \cap \mathbb{Z} \\ p_2 & t \in I_2 \cap \mathbb{Z} \\ 0 & \text{otherwise.} \end{cases}$$

Let Δ be the partition of the form $I = I_1 + I_2$. It is easy to obtain that

$$m_1 = \sqrt{\frac{1}{2}h_1 + \frac{h_2 p_2}{p_1}}, \quad m_2 = \sqrt{\frac{1}{2}h_2}$$

and the inequality $m_1^2 < m_2^2$ is equivalent to $\frac{h_1}{h_2} < 1 - \frac{2}{(\frac{p_1}{p_2})}$. So, the condition that Δ is an effective partition is equivalent to

$$\frac{h_1}{h_2} < 1 - \frac{2}{(\frac{p_1}{p_2})}. \qquad (\text{A}\cdot 1)$$

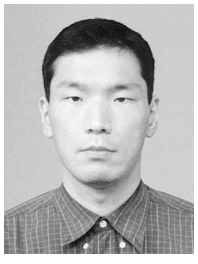
Appendix D: Comparison of Kangaroo and BSGS

In [7], Teske explicitly analyses the running time of kangaroo method for the interval-$[a, b]$-DLP, which is the discrete log problem with the solutions in the interval $[a, b]$. Here, as we mentioned in Sect. 1, we remark that kangaroo method can be employed for computing the order of a finite group as well as computing the DLP, and we can also see that the efficiency of kangaroo method for computing the order of a finite group

is as comparable as that for computing the DLP. According to [7], the expected running time of kangaroo method is given by $3.3\sqrt{b-a}$ (less-storage case, page 285 in [7]) or $2\sqrt{b-a}$ (more-storage case, page 286 in [7]). In [9], Teske and Stein also analyse the running time of BSGS for the interval $[a, b]$-order counting problem. According to [9, §2.1], the worst running time of BSGS is given by $2\sqrt{b-a}$. So, generally speaking, if we can use sufficiently large storage, it is expected to be that the BSGS needs smaller number of group operations than the kangaroo method.

Koh-ichi Nagao received the Doctor of Mathematical Science degree from Kyushu University, Fukuoka, Japan, in 1997. He has been an assistant professor in the Faculty of Engineering, Kanto Gakuin University. Dr. Nagao is a member of the Mathematical Society of Japan and the Japan Society for Industrial and Applied Mathematics.

Shigenori Uchiyama received the B.S., M.S. and Ph.D. degrees from Kyushu University, Fukuoka, Japan, in 1991, 1993 and 1996, respectively. Since joining NTT Laboratories in 1996, he has been engaged in research on cryptography and information security. In 2000 – 2001, he was a visiting scholar of the Computer Science Department at the University of Southern Californina. He is currently a research scientist of NTT Information Sharing Platform Laboratories. Dr. Uchiyama is a member of the Mathematical Society of Japan and the Japan Society for Industrial and Applied Mathematics.



Naoki Kanayama received his B.E., B.S., M.S. and Ph.D. degrees from Waseda University, Tokyo, Japan, in 1994, 1996, 1998 and 2003 respectively. He is a research fellow of the Japan Society for the Promotion of Science. Dr. Kanayama is a member of the Mathematical Society of Japan.

Kazuto Matsuo received the B.E., M.E., and D.E. from Chuo University, Tokyo, Japan in 1986, 1988, and 2001, respectively. He joined Toyo Communication Equipment Co., LTD from 1988 to 2001. He is currently a professor in the Research and Development Initiative at Chuo University. His current research interests include cryptography.