A Novel Multi Exponentiation Method

Raveen R. Goundar *

Abstract—**The efficiency and security of most elliptic curve cryptosystems are based on multi exponentiation, such as the verification process in elliptic curve digital signature algorithm. Simultaneous methods are considered to be the most efficient for multi exponentiation. In this paper, we propose a method to construct an addition chain for simultaneous multi exponentiation, which has never been considered in the literature before. We discuss the strategy for the two dimension case, $b_1{}^e b_2{}^f$ and assume that such strategy could be generalize for arbitrary cases.**

Keywords: Addition chain, Fibonacci sequence, golden ratio, elliptic curve cryptosystems.

1 Introduction

Most elliptic curve cryptosystems [6, 8] require computations of several exponentials with distinct bases and distinct exponents which are termed as multi exponentiation. The computation of each exponentiation separately and multiplying the result at the end could be very costly. In such case, simultaneous methods are considered to be more efficient for multi exponentiation. The conventional simultaneous multi exponentiation method is known as Shamir's trick [1]. It is dependable on binary representation of exponents and utilizes square-and-multiply method, hence susceptible to simple power analysis attack (SPA), recently discovered by kocher [7]. It has been suggested by Byrne et al. [2] that the use of doubling-free addition chain can provide resistant to SPA attack, hence we will consider it in our case. Note that addition chain involving one doubling that is numeral 2 is unaffected by SPA attack since it is a necessary computation in almost all exponentiations in elliptic curve cryptosystems. The use of addition chain for simultaneous multi exponentiation has never been considered in the literature before. Conventionally, addition chains are not suitable for variable exponents and fixed bases but only suitable for fixed exponent and variable bases [9, 5]. In this paper, we will propose a novel method for simultaneous multi exponentiation using addition chain.

We will extend the previously proposed golden ratio addition-subtraction chain (GRASC) method [3] to suit the computation of multi exponentiation for our purpose. We will term it as simultaneous golden ratio addition chain method or SGRAC method in short. We will consider computation of two dimension case, that is $b_1{}^e b_2{}^f$ where b_1, b_2 are element of an abelian group and $e, f \in \mathbb{Z}$. We assume that such strategy could be generalize for arbitrary cases.

The rest of this paper is organized as follows. In section 2, we give brief overview on addition chains and Fibonacci sequence. We also review the GRASC method. In section 3, we discuss the SGRAC method and state its algorithm. Later we propose a multi exponentiation algorithm based on SGRAC method. In section 4, we discuss our experimental results.

2 Background

In this section, we briefly state some classic definitions used in the study of addition chains and an overview on Fibonacci sequence. More details could be cited from [1, 4].

Definition 1 *An addition chain computing an integer k is given by two sequences $v = (v_0, \ldots, v_\ell)$ and $w = (w_1, \ldots, w_\ell)$ such that $v_0 = 1, v_\ell = k, v_i = v_r + v_s$, for all $1 \leq i \leq \ell$ with respect to $w_i = (r, s)$ and $0 \leq r, s \leq i - 1$. The length of the addition chain is ℓ.*

Definition 2 *An addition-subtraction chain is similar to an addition chain except that the coordinate $v_i = v_r + v_s$ is replaced by $v_i = v_r + v_s$ or $v_i = v_r - v_s$.*

Definition 3 *The Fibonacci sequence is defined as $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$ where $F_0 = 0$ and $F_1 = 1$.*

The Fibonacci sequence has many properties [4, 10] we recall one here, by stating the following Binet's Formula.

Theorem 1 *Binet's Formula:*

$$F_n = \frac{\phi^n - (1-\phi)^n}{\sqrt{5}}, \quad \forall n \in \mathbb{N},$$

where $\phi = \frac{1+\sqrt{5}}{2}$ is the positive root of the real polynomial $X^2 - X - 1$.

*Graduate School of Mathematics and Information Science, Kochi University, Japan. Email: raveenrg@is.kochi-u.ac.jp
Manuscript Submitted: October 9, 2007.

(Revised online publication: 17 June 2008)

From the above theorem it is easy to deduce the following classical result.

$$\lim_{n\to\infty} \frac{F_n}{F_{n-1}} = \phi\,, \tag{1}$$

where ϕ is a golden ratio, also called a golden section.

2.1 Review on GRASC Method [3]

Here we review the GRASC method for finding doubling-free short addition-subtraction chain by utilizing a precise golden ratio.

The GRASC method considers making chain starting from the last term, which is the input k and aims to follow a Fibonacci pattern using the fact from equation (1). Hence, it tries to maintain a near golden ratio value between two succeeding terms. It begins by letting

$$\begin{aligned} u_0 &= k\,, \\ u_1 &= [u_0 \times \phi^{-1}]\,, \\ u_i &= u_{i-2} - u_{i-1} \text{for } i = 2, 3, \ldots \end{aligned} \tag{2}$$

Here u_i denotes the reverse of v_i that is, $u_i = v_{\ell-i}$. If continued with the procedure (2), u_i will exponentially deviate from $(u_{i-1} \times \phi^{-1})$ as i increases. In order to overcome this problem, a parameter MAXIMALGAP is introduced, such that the above procedure (2) terminates whenever

$$|u_i - (u_{i-1} \times \phi^{-1})| > \text{ MAXIMALGAP or } u_i \leqslant \frac{u_{i-1}}{2}\,.$$

In such case, a new u_i is defined to be the nearest integer of $(u_{i-1} \times \phi^{-1})$. Then procedure (2) is resumed with u_{i-1} and new u_i as the initial terms. The old u_i is included in the chain between u_{i-1} and new u_i, as a consequence there is a gap $g_j = |\text{old } u_i - \text{new } u_i|$, which is included in the storage. Note that, subtraction is involved whenever old $u_i <$ new u_i. Another parameter LOWERBOUND is introduced to cease the procedure (2) when $u_i \leq$ LOWERBOUND. The storage initially consists of $1, 2$, and 3. Later g_j's are included in the storage. Once the execution of procedure (2) is ceased, the last two u_i's of the chain is included in the storage. Thus, using the storage, a short addition chain is found randomly without using doubling, except for numeral 2. Finally, this chain is joined to the third last u_i of the previous chain resulting in a moderately short addition-subtraction chain for the given input k. Note that the storage capacity is dependent on the experimentally selected values of the two parameters.

3 Simultaneous golden ratio addition chain method (SGRAC method)

In the case of multi exponentiation, such as computation of $b_1^{e_0} b_2^{f_0}$, we shall follow the GRASC method with $u_i = e_0 x_1 + f_0 x_2$ as an input. We pair the exponents with variables x_1 and x_2 to distinguish between them. We select a suitable parameters, MAXIMALGAP and LOWERBOUND of the form $e_i x_1 + f_i x_2$. We compare the coefficients of each variables while checking the conditions given by the parameters. If either of the coefficients does not satisfy the condition, we take the required actions as mentioned in GRASC method. Note that SGRAC method will include negative numbers as the storage, that is a gap $g_j =$(old $u_i -$new u_i) is included in the storage contrary to GRASC method which only includes absolute values of g_j's. Hence, the exponentiation based on SGRAC method involves addition (multiplication) throughout the process. The Algorithm 1 explains the SGRAC method in detail.

Algorithm 1 SGRAC method

Input: Exponents e_0 and f_0.
Output: $w = \{w_0, \ldots, w_\ell\}$, $G = \{g_1, \ldots, g_j, s_1, s_2\}$, SAC_{x_1}, SAC_{x_2} and m.

1. MAXIMALGAP $\leftarrow$ choose a suitable parameter
2. LOWERBOUND $\leftarrow$ choose a suitable parameter
3. $\phi^{-1} \leftarrow \frac{-1+\sqrt{5}}{2}$
4. $u_0 \leftarrow e_0 x_1 + f_0 x_2$ (x_1 and x_2 are variables)
5. $u_1 \leftarrow [u_0 \times \phi^{-1}]$
6. $u_2 \leftarrow u_0 - u_1$
7. $v = \{u_0, u_1, u_2\}$
8. $S = \{1x_1, 1x_2, 2x_1, 2x_2, 3x_1, 3x_2\}$
9. $m = \{0, 0\}$
10. $G = \emptyset$
11. $i \leftarrow 2$
12. $j \leftarrow 1$
13. **while** $u_i >$ LOWERBOUND **do**
14. **if** $|u_i - (u_{i-1} \times \phi^{-1})| >$ MG or $u_i \leqslant \frac{u_{i-1}}{2}$ **then**
15. $m_i = 1\,, m_{i+1} = 0$
16. $m \leftarrow m \cup \{m_i\,, m_{i+1}\}$
17. $i \leftarrow i + 1$
18. $u_i \leftarrow [u_{i-2} \times \phi^{-1}]$
19. $v \leftarrow v \cup \{u_i\}$
20. $u_{i+1} \leftarrow u_{i-2} - u_i$
21. $v \leftarrow v \cup \{u_{i+1}\}$
22. $g_j \leftarrow (u_i - u_{i+1})$
23. $G \leftarrow G \cup \{g_j\}$
24. $j \leftarrow j + 1$
25. $i \leftarrow i + 1$
26. **else**
27. $m_i = 0$
28. $m \leftarrow m \cup \{\, m_i\}$
29. $i \leftarrow i + 1$
30. $u_i \leftarrow u_{i-2} - u_{i-1}$
31. $v \leftarrow v \cup \{u_i\}$
32. $S \leftarrow S \cup \{u_{i-1}, u_i\}$
33. $m_i = 2\,, m_{i+1} = 2$
34. $m \leftarrow m \cup \{m_i, m_{i+1}\}$
35. $max \leftarrow j$
36. **if** $e_i >$ LOWERBOUND **then**
37. $S \leftarrow S \cup \{f_{i-1} x_2\}$
38. $s_1 \leftarrow f_i x_2\,,\ s_2 \leftarrow (f_{i-1} x_2 - f_i x_2)$
39. $G \leftarrow G \cup \{s_1\,, s_2\}$
40. $SAC_{x_2} \leftarrow$ Short addition chain using absolute values of all x_2 terms in G and S
41. $j \leftarrow max + 1$
42. Repeat step 13 to step 32 for the x_1 terms only
43. $SAC_{x_1} \leftarrow$ Short addition chain using absolute values of all x_1 terms in G and S
44. $v \leftarrow v \cup SAC_{x_1}$
45. All m_i's are zero in SAC_{x_1}

(Revised online publication: 17 June 2008)

46. **else**
47. $S \leftarrow S \cup \{e_{i-1}x_1\}$
48. $s_1 \leftarrow e_i x_1$, $s_2 \leftarrow (e_{i-1}x_1 - e_i x_1)$
49. $G \leftarrow G \cup \{s_1, s_2\}$
50. $SAC_{x_1} \leftarrow$ Short addition chain using absolute values of all x_1 terms in G and S
51. $j \leftarrow max + 1$
52. Repeat step 13 to step 32 for the x_2 terms only
53. $SAC_{x_2} \leftarrow$ Short addition chain using absolute values of all x_2 terms in G and S
54. $v \leftarrow v \cup SAC_{x_2}$
55. All m_i's are zero in SAC_{x_2}
56. $G \leftarrow$ reverse the elements g_j's in G and rename it in increasing order starting from numeral 1
57. $v \leftarrow$ reverse the arrangements of elements in v and rename in increasing order starting from v_0 to v_ℓ
58. $m \leftarrow$ reverse the arrangements of elements in m and rename in increasing order starting from m_0 to m_ℓ
59. $w \leftarrow$ is a set containing index for the computation of each v_i in v
60. **return** $w = \{w_0, \ldots, w_\ell\}$, $G = \{g_1, \ldots, g_j, s_1, s_2\}$, SAC_{x_1}, SAC_{x_2} and m

Note that in Algorithm 1, $u_i = e_i x_1 + f_i x_2$ and the symbol MG represents the MAXIMALGAP.

Example 1. Evaluate Algorithm 1 for the input exponents $e_0 = 90287$ and $f_0 = 1835008$.

In order to differentiate between the two exponents values, we introduce two variables x_1 and x_2. We experimentally selected suitable values for the LOWERBOUND and the MAXIMALGAP to be $10x_1 + 10x_2$ and $6x_1 + 6x_2$, respectively. Note that we will be comparing the coefficients of either x_1 or x_2 while checking the conditions using the two parameters. In the following, we will denote $u_i = e_i x_1 + f_i x_2$ for $i \geq 0$. We begin by letting

$$\begin{aligned}
u_0 &= e_0x_1 + f_0x_2 = 90287x_1 + 1835008x_2\,, && m_0 = 0\\
u_1 &= [u_0 \times \phi^{-1}] = 55800x_1 + 1134097x_2\,, && m_1 = 0\\
u_2 &= u_0 - u_1 = 34487x_1 + 700911x_2\,, && m_2 = 0\\
u_3 &= u_1 - u_2 = 21313x_1 + 433186x_2\,, && m_3 = 0\\
u_4 &= u_2 - u_3 = 13174x_1 + 267725x_2\,, && m_4 = 0\\
u_5 &= u_3 - u_4 = 8139x_1 + 165461x_2\,, && m_5 = 0\\
u_6 &= u_4 - u_5 = 5035x_1 + 102264x_2\,, && m_6 = 0\\
u_7 &= u_5 - u_6 = 3104x_1 + 63197x_2\,, && m_7 = 1
\end{aligned}$$

since u_7 does not satisfy the condition $|u_7 - (u_6 \times \phi^{-1})| < 6x_1 + 6x_2$, therefore we utilize an inverse of a golden ratio to get the next u_i. That is

$$u_8 = [u_6 \times \phi^{-1}] = 3112x_1 + 63203x_2\,. \qquad m_8 = 0$$

There exist a gap, $g_1 = u_7 - u_8 = -8x_1 - 6x_2$, which we include in the storage S. We continue by letting

$$\begin{aligned}
u_9 &= u_6 - u_8 = 1923x_1 + 39061x_2\,, && m_9 = 0\\
u_{10} &= u_8 - u_9 = 1189x_1 + 24142x_2\,, && m_{10} = 0\\
u_{11} &= u_9 - u_{10} = 734x_1 + 14919x_2\,, && m_{11} = 0\\
u_{12} &= u_{10} - u_{11} = 455x_1 + 9223x_2\,, && m_{12} = 0\\
u_{13} &= u_{11} - u_{12} = 279x_1 + 5696x_2\,, && m_{13} = 0\\
u_{14} &= u_{12} - u_{13} = 176x_1 + 3527x_2\,, && m_{14} = 1
\end{aligned}$$

since u_{14} does not satisfy the condition $|u_{14} - (u_{13} \times \phi^{-1})| < 6x_1 + 6x_2$, therefore we utilize an inverse of a golden ratio to get the next u_i. That is

$$u_{15} = [u_{13} \times \phi^{-1}] = 172x_1 + 3520x_2. \qquad m_{15} = 0$$

There exist a gap, $g_2 = u_{14} - u_{15} = 4x_1 + 7x_2$, which we include in the storage S. We continue by letting

$$\begin{aligned}
u_{16} &= u_{13} - u_{15} = 107x_1 + 2176x_2\,, && m_{16} = 0\\
u_{17} &= u_{15} - u_{16} = 65x_1 + 1344x_2\,, && m_{17} = 0\\
u_{18} &= u_{16} - u_{17} = 42x_1 + 832x_2\,, && m_{18} = 0\\
u_{19} &= u_{17} - u_{18} = 23x_1 + 512x_2\,, && m_{19} = 0\\
u_{20} &= u_{18} - u_{19} = 19x_1 + 320x_2\,, && m_{20} = 2\\
u_{21} &= u_{19} - u_{20} = 4x_1 + 192x_2\,. && m_{21} = 2
\end{aligned}$$

We stop the above continuous procedure at u_{21}, since the coefficient of x_1 in u_{21} transcends the coefficient of x_1 in the LOWERBOUND. We include $e_{20}x_1 = 19x_1$ in the storage S.

We let s_1 denote $e_{21}x_1 = 4x_1$ and s_2 denote $e_{20}x_1 - e_{21}x_1 = 15x_1$. We include s_1 and s_2 in G. Henceforth, we will only consider x_2 terms and resume the above process. We let

$$u_{22} = f_{20}x_2 - f_{21}x_2 = 128x_2\,, \qquad m_{22} = 1$$

since u_{22} does not satisfy the condition $|f_{22}x_2 - (f_{21}x_2 \times \phi^{-1})| < 6x_2$, therefore we utilize an inverse of a golden ratio to get the next u_i. That is

$$u_{23} = [f_{21}x_2 \times \phi^{-1}] = 119x_2. \qquad m_{23} = 0$$

There exist a gap, $g_3 = u_{22} - u_{23} = 9x_2$, which we include in the storage S. We continue by letting

$$\begin{aligned}
u_{24} &= f_{21}x_2 - u_{23} = 73x_2\,, && m_{24} = 0\\
u_{25} &= u_{23} - u_{24} = 46x_2\,, && m_{25} = 0\\
u_{26} &= u_{24} - u_{25} = 27x_2\,, && m_{26} = 0\\
u_{27} &= u_{25} - u_{26} = 19x_2\,, && m_{27} = 0\\
u_{28} &= u_{26} - u_{27} = 8x_2\,. && m_{28} = 0
\end{aligned}$$

We stop the above process at u_{28}, since it transcends the given LOWERBOUND$= 10x_2$. Hence, we include the last two x_2 terms, that is $19x_2$ and $8x_2$, in the storage S.

We have $G = \{g_1 = -8x_1 - 6x_2, g_2 = 4x_1 + 7x_2, g_3 = 9x_2, s_1 = 4x_1, s_2 = 15x_1\}$. Now we reverse the arrangements of g_j's in G and rename it in the increasing order starting with numeral 1. Hence, we have $G = \{g_1 = 9x_2, g_2 = 4x_1 + 7x_2, g_3 = -8x_1 - 6x_2, s_1 = 4x_1, s_2 = 15x_1\}$. Considering the storage S, we have $S = \{1x_1, 1x_2, 2x_1, 2x_2, 3x_1, 3x_2, 19x_1, 19x_2, 8x_2\}$. Now, using absolute values of all elements in G and S, we shall construct a short addition chain for x_1 terms and x_2 terms, separately.

First, we shall consider constructing doubling-free short addition chain including absolute values of all x_1 terms in G and S. We will denote it as SAC_{x_1}. Hence, we have

$$\{1x_1, 2x_1, 3x_1, 19x_1, 4x_1, 8x_1, 4x_1, 15x_1\}.$$

Excluding the repeated terms and rearrangement results

$$\{1x_1, 2x_1, 3x_1, 4x_1, 8x_1, 15x_1, 19x_1\}.$$

It follows that $1x_1 + 2x_1 \rightarrow 3x_1$, $1x_1 + 3x_1 \rightarrow 4x_1$, $2x_1 + 3x_1 \rightarrow 5x_1$, $3x_1 + 5x_1 \rightarrow 8x_1$, $2x_1 + 8x_1 \rightarrow 10x_1$,

(Revised online publication: 17 June 2008)

$5x_1 + 10x_1 \to 15x_1$ and $4x_1 + 15x_1 \to 19x_1$. Hence, the following results the SAC_{x_1}.

$$1 \to 2 \to 3 \to 4 \to 5 \to 8 \to 10 \to 15 \to 19\,.$$

Next, we shall consider constructing doubling-free short addition chain including absolute values of all x_2 terms in G and S. We will denote it as SAC_{x_2}. Hence, we have

$$\{1x_2, 2x_2, 3x_2, 19x_2, 8x_2, 9x_2, 7x_2, 6x_2\}.$$

Excluding the repeated terms and rearrangement results

$$\{1x_2, 2x_2, 3x_2, 6x_2, 7x_2, 8x_2, 9x_2, 19x_2\}.$$

It follows that $1x_2 + 2x_2 \to 3x_2$, $2x_2 + 3x_2 \to 5x_2$, $1x_2 + 5x_2 \to 6x_2$, $1x_2 + 6x_2 \to 7x_2$, $1x_2 + 7x_2 \to 8x_2$, $1x_2 + 8x_2 \to 9x_2$, $2x_2 + 9x_2 \to 11x_2$ and $8x_2 + 11x_2 \to 19x_2$. Hence the following results the SAC_{x_2}.

$$1 \to 2 \to 3 \to 5 \to 6 \to 7 \to 8 \to 9 \to 11 \to 19\,.$$

Finally, we join SAC_{x_2} at u_{26} to complete the overall chain for $e_0x_1 + f_0x_2$. Note that the old u_{27} and old u_{28} will be replaced with the ones in SAC_2. The continuation of the chain will be denoted as $u_{27} = 19x_2$, $u_{28} = 11x_2$, $u_{29} = 9x_2$, $u_{30} = 8x_2$, $u_{31} = 7x_2$, $u_{32} = 6x_2$, $u_{33} = 5x_2$, $u_{34} = 3x_2$, $u_{35} = 2x_2$, $u_{36} = 1x_2$. Note that $m_i = 0$ for $i = 27, \ldots, 36$.

We reverse the arrangements of m_i's and rename it in the increasing order starting from 0 to 36. That is $m = \{0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0\}$.
The set v contains all the u_i's of the above chain where $v = \{u_0, u_1, \ldots, u_{36}\}$. We reverse the arrangements of elements in v and rename it in the increasing order starting from v_0 to v_{36}. Hence, we have $v = \{v_0, v_1, \ldots, v_{36}\}$ where $v_i = u_{\ell-i}$ for all i.

Next, we let either $w_i = (j,k)$ or $w_i = (k)$ to denote the index of v_i's involved in the computation of each v_i in v. This could be found from the above chain. Hence, we have $w = \{w_0, w_1 = (0,0), w_2 = (0,1), \ldots, w_{36} = (34,35)\}$.

Algorithm 2 Multi-exponentiation using SGRAC method

Input: b_1, b_2 are elements of an abelian group and e_0, $f_0 \in \mathbb{Z}$.
Output: The element $b_1^{e_0} b_2^{f_0}$.

Precomputation (SGRAC method)
1. $w = \{w_0, \ldots, w_\ell\}$, $G = \{g_1, \ldots, g_j, s_1, s_2\}$, SAC_{x_1}, SAC_{x_2} and m
2. $b_1 \leftarrow x_1$, $b_2 \leftarrow x_2$
3. Compute all elements in G using SAC_{x_1} and SAC_{x_2}, store it in G

Main loop
4. **if** (step 44 of Alg. 1 holds) **then**
5. $v_0 \leftarrow b_1^1$
6. **else**
7. $v_0 \leftarrow b_2^1$
8. $j \leftarrow 1$
9. $r \leftarrow 1$
10. **for** $i = 1$ **to** ℓ
11. **if** $m_i = 0$ **then**
12. $v_i \leftarrow v_j \times v_k$ $[w_i = (j,k)]$
13. **else if** $m_i = 1$ **then**
14. $v_i \leftarrow g_j \times v_k$ $[w_i = (k)]$
15. $j \leftarrow j + 1$
16. **else**
17. $v_i \leftarrow v_j \times v_k \times s_r$ $[w_i = (j,k)]$
18. $r \leftarrow r + 1$
19. **return** v_ℓ

Example 2. Evaluate Algorithm 2 for the inputs $e_0 = 90287$, $f_0 = 1835008$ and b_1, b_2 as elements of an abelian group.

Precomputation: (SGRAC method)

1. We have $w = \{w_0, w_1 = (0,0), w_2 = (0,1), \ldots, w_{36} = (34,35)\}$, G, SAC_{x_1}, SAC_{x_2} and m from Example 1.
2. We replace x_1 by b_1 and x_2 by b_2 for all elements in G.
3.Then using SAC_{x_1} and SAC_{x_2}, we compute all the elements in G and again store it in G.
Hence, we get $G = \{g_1 = b_2^9, g_2 = b_1^4 b_2^7, g_3 = b_1^{-8} b_2^{-6}, s_1 = b_1^4, s_2 = b_1^{15}\}$.
The following results the evaluation stage. The step 44 of Algorithm 1 does not hold, therefore $v_0 = {b_2}^1$. It follows that

$$m_0 = 0\,, \quad w_0 = (-)\,, \quad v_0 = {b_2}^1\,,$$

$$m_1 = 0\,, \quad w_1 = (0,0)\,, \quad v_1 = v_0 \times v_0 = {b_2}^2\,,$$

$$m_2 = 0\,, \quad w_2 = (0,1)\,, \quad v_2 = v_0 \times v_1 = {b_2}^3\,,$$

$$m_3 = 0\,, \quad w_3 = (1,2)\,, \quad v_3 = v_1 \times v_2 = {b_2}^5\,,$$

$$m_4 = 0\,, \quad w_4 = (0,3)\,, \quad v_4 = v_0 \times v_3 = {b_2}^6\,,$$

$$m_5 = 0\,, \quad w_5 = (0,4)\,, \quad v_5 = v_0 \times v_4 = {b_2}^7\,,$$

$$m_6 = 0\,, \quad w_6 = (0,5)\,, \quad v_6 = v_0 \times v_5 = {b_2}^8\,,$$

$$m_7 = 0\,, \quad w_7 = (0,6)\,, \quad v_7 = v_0 \times v_6 = {b_2}^9\,,$$

$$m_8 = 0\,, \quad w_8 = (1,7)\,, \quad v_8 = v_1 \times v_7 = {b_2}^{11}\,,$$

$$m_9 = 0\,, \quad w_9 = (6,8)\,, \quad v_9 = v_6 \times v_8 = {b_2}^{19}\,,$$

(Revised online publication: 17 June 2008)

$$m_{10}=0\,,\quad w_{10}=(6,9)\,,\qquad v_{10}=v_6\times v_9={b_2}^{27}\,,$$

$$m_{11}=0\,,\quad w_{11}=(9,10)\,,\qquad v_{11}=v_9\times v_{10}={b_2}^{46}\,,$$

$$m_{12}=0\,,\quad w_{12}=(10,11)\,,\quad v_{12}=v_{10}\times v_{11}={b_2}^{73}\,,$$

$$m_{13}=0\,,\quad w_{13}=(11,12)\,,\quad v_{13}=v_{11}\times v_{12}={b_2}^{119}\,,$$

$$m_{14}=1\,,\quad w_{14}=(13)\,,\qquad v_{14}=g_1\times v_{13}={b_2}^{128}\,,$$

$$m_{15}=2\,,\quad w_{15}=(12,13)\,,\quad v_{15}=v_{12}\times v_{13}\times s_1={b_1}^{4}{b_2}^{192}\,,$$

$$m_{16}=2\,,\quad w_{16}=(14,15)\,,\quad v_{16}=v_{14}\times v_{15}\times s_2={b_1}^{19}{b_2}^{320}\,,$$

$$m_{17}=0\,,\quad w_{17}=(15,16)\,,\quad v_{17}=v_{15}\times v_{16}={b_1}^{23}{b_2}^{512}\,,$$

$$m_{18}=0\,,\quad w_{18}=(16,17)\,,\quad v_{18}=v_{16}\times v_{17}={b_1}^{42}{b_2}^{832}\,,$$

$$m_{19}=0\,,\quad w_{19}=(17,18)\,,\quad v_{19}=v_{17}\times v_{18}={b_1}^{65}{b_2}^{1344}\,,$$

$$m_{20}=0\,,\quad w_{20}=(18,19)\,,\quad v_{20}=v_{18}\times v_{19}={b_1}^{107}{b_2}^{2176}\,,$$

$$m_{21}=0\,,\quad w_{21}=(19,20)\,,\quad v_{21}=v_{19}\times v_{20}={b_1}^{172}{b_2}^{3520}\,,$$

$$m_{22}=1\,,\quad w_{22}=(21)\,,\qquad v_{22}=g_2\times v_{21}={b_1}^{176}{b_2}^{3527}\,,$$

$$m_{23}=0\,,\quad w_{23}=(20,21)\,,\quad v_{23}=v_{20}\times v_{21}={b_1}^{279}{b_2}^{5696}\,,$$

$$m_{24}=0\,,\quad w_{24}=22,23)\,,\quad v_{24}=v_{22}\times v_{23}={b_1}^{455}{b_2}^{9223}\,,$$

$$m_{25}=0\,,\quad w_{25}=(23,24)\,,\quad v_{25}=v_{23}\times v_{24}={b_1}^{734}{b_2}^{14919}\,,$$

$$m_{26}=0\,,\quad w_{26}=(24,25)\,,\quad v_{26}=v_{24}\times v_{25}={b_1}^{1189}{b_2}^{24142}\,,$$

$$m_{27}=0\,,\quad w_{27}=(25,26)\,,\quad v_{27}=v_{25}\times v_{26}={b_1}^{1923}{b_2}^{39061}\,,$$

$$m_{28}=0\,,\quad w_{28}=(26,27)\,,\quad v_{28}=v_{26}\times v_{27}={b_1}^{3112}{b_2}^{63203}\,,$$

$$m_{29}=1\,,\quad w_{29}=(28)\,,\qquad v_{29}=g_3\times v_{28}={b_1}^{3104}{b_2}^{63197}\,,$$

$$m_{30}=0\,,\quad w_{30}=(27,28)\,,\quad v_{30}=v_{27}\times v_{28}={b_1}^{5035}{b_2}^{102264}\,,$$

$$m_{31}=0\,,\quad w_{31}=(29,30)\,,\quad v_{31}=v_{29}\times v_{30}={b_1}^{8139}{b_2}^{165461}\,,$$

$$m_{32}=0\,,\quad w_{32}=(30,31)\,,\quad v_{32}=v_{30}\times v_{31}={b_1}^{13174}{b_2}^{267725}\,,$$

$$m_{33}=0\,,\quad w_{33}=(31,32)\,,\quad v_{33}=v_{31}\times v_{32}={b_1}^{21313}{b_2}^{433186}\,,$$

$$m_{34}=0\,,\quad w_{34}=(32,33)\,,\quad v_{34}=v_{32}\times v_{33}={b_1}^{34487}{b_2}^{700911}\,,$$

$$m_{35}=0\,,\quad w_{35}=(33,34)\,,\quad v_{35}=v_{33}\times v_{34}={b_1}^{55800}{b_2}^{1134097}\,,$$

$$m_{36}=0\,,\quad w_{36}=(34,35)\,,\quad v_{36}=v_{34}\times v_{35}={b_1}^{90287}{b_2}^{1835008}\,.$$

(Revised online publication: 17 June 2008)

Table 1: The distribution of chains for 1000 randomly selected integers e and f of 160 bit.

SGRAC length (ℓ)	# inputs k
SGRAC-291	1
SGRAC-292	1
SGRAC-293	0
SGRAC-294	3
SGRAC-295	5
SGRAC-296	13
SGRAC-297	25
SGRAC-298	66
SGRAC-299	103
SGRAC-300	139
SGRAC-301	215
SGRAC-302	177
SGRAC-303	152
SGRAC-304	70
SGRAC-305	26
SGRAC-306	4

Table 2: The distribution of MAXIMALGAP for 1000 randomly selected integers e and f of 160 bit.

MAXIMALGAP	5	6	7	8	9	10
# inputs k	51	98	100	81	137	93

MAXIMALGAP	11	12	13	14	15
# inputs k	103	88	75	100	74

Table 3: The distribution of LOWERBOUND for 1000 randomly selected integers e and f of 160 bit.

LOWERBOUND	4	5	6	7	8	9	10
# inputs k	115	77	102	87	86	79	71

LOWERBOUND	11	12	13	14	15	16	17
# inputs k	61	66	54	36	39	35	12

LOWERBOUND	18	19	20	21	22
# inputs k	23	14	15	20	8

Table 4: The distribution of storage for 1000 randomly selected integers e and f of 160 bit.

Storage capacity	16	17	18	19	20	21	22
# inputs k	60	59	33	7	1	1	15

Storage capacity	23	24	25	26	27	28	29
# inputs k	33	35	50	45	42	55	55

Storage capacity	30	31	32	33	34	35	36
# inputs k	68	71	71	79	89	70	61

4 Experimental Results

In this section, the experimental data due to K. Shiota, shows the factors necessitated in obtaining the best results, that is the least chain length.

We carried out an experiment to analyze the two dimension multi exponentiation based on SGRAC method using a python programming language on 1.66 GHz Intel Core Duo processor. We randomly selected 1000 integers e and f of 160 bits and set the searching range of the parameters, LOWERBOUND to be between $2x_1 + 2x_2$ to $22x_1+22x_2$ and MAXIMALGAP to be between $5x_1+5x_2$ to $15x_1 + 15x_2$. It took 209 trials to obtain chains of lengths between 291 to 306 as shown on Table 1. On average it took about 2.89 seconds to find each chain. The Table 2 , 3 and 4 shows the distribution of MAXIMALGAP, LOWERBOUND and the storage that gives the least chain length, respectively. The average chain length is found to be 301 and the average storage capacity is found to be 26. The conventional Shamir's trick uses 160 squaring operations and 82 multiplication operation for two dimension case, where as multi exponentiation based on SGRAC method utilizes 301 multiplication (best case). Hence, in comparison to Shamir's trick method, the SGRAC method seems to be costly. However for further work, reducing the length of the addition chain (SGRAC) can enhance the efficiency of the proposed multi exponentiation.

5 Conclusion

In this paper we have proposed a novel method for the simultaneous computation of multi exponentiation, that is by employing addition chains. The experimental results for the two dimension case shows that SGRAC method gives an average chain of length 301 with average storage capacity of 26. Further work may include in reducing the chain length and the storage capacity in order to enhance further efficiency. We assume that SGRAC method could be further generalize to arbitrary cases.

References

[1] Avanzi, R.M., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., and Vercauteren, F., *Handbook of Elliptic and Hyperelliptic Curve Cryptography,* CRC Press, 2005.

[2] Byrne, A., Meloni, N., Crowe, F., Marnane, W.P., Tisserand, A., and Popovici, E.M., "SPA resistant Elliptic Curve Cryptosystem using Addition Chains", *International Conference on Information Technology-ITNG'07*, pp.995-1000, 2007.

[3] Goundar, R.R., Shiota, K., and Toyonaga, M., "New Strategy for Doubling-free Short Addition-

Subtraction Chain", *International Journal of Computational and Applied Mathematics*, To appear.

[4] Knuth, D., "Fundamental Algorithms", *The Art of Computer Programming*, volume 1, Addision-Wesley,(1981).

[5] Kobayashi, K., Morita, H., and Hakuta, M., "Multi Scalar-Multiplication Algorithm over Elliptic Curve", *IEICE Transactions on Information and Systems*, E84-D, No.2, pp.271-276, 2001.

[6] Koblitz, N., "Elliptic curve cryptosystems", *Mathematics of Computation*, 48(177):203-209, Jan. 1987.

[7] Kocher, P., Jaffe, J., and Jun, B., "Differential power analysis", volume 1666 of *Lecture Notes in Computing Science*, pages 388-397. Springer-Verlag, 1999.

[8] Miller, V.S., "Uses of elliptic curves in cryptography", In H.C. Williams, editor, *Advances in Cryptology-CRYPTO'85*, volume 218 of *Lecture Notes in Computing Science*, pages 417-428. Springer-Verglag, 1986.

[9] Menezes, A.J., vanOorschot, P.C., and Vanstone, S.A., *Handbook of Applied Cryptography,* CRC Press, 1997.

[10] Vorobiev, N., *Fibonacci Numbers.* Birkhuser Verlag, 2002.

(Revised online publication: 17 June 2008)