A Hybrid Addition Chain Method For Faster Scalar Multiplication

M A MOHAMED
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin, Besut, 22200 MALAYSIA
mafendee@unisza.edu.my
M R MD SAID
Institute for Mathematical Research
Universiti Putra Malaysia, Serdang, 43400 MALAYSIA
mrushdan@upm.edu.my

Abstract: Solutions to addition chain problem can be applied to operations involving huge number such as scalar multiplication in elliptic curve cryptography. Recently, a decomposition method was introduced, with an intention to generate addition chain with minimal possible terms. Totally different from others, this new method uses rule representation for prime factors of n, and a new algorithm to generate a complete chain for n. Although the chain is not always optimal, the method is shown to outclass other existing methods for certain cases of n. The method is based on prime power decomposition and it can be seen as a two-layered approach, prime layer and prime power layer. In this paper, we adapt an idea of non-adjacent form into decomposition method at prime layer. This new hybrid method is called signed decomposition method. Our objective is to reduce the number of addition operations for each p by transforming an original unsigned rule into a signed rule. The study shows that the length of this new chain is confined to the same boundary as that of an optimal chain. A series of tests shows that our method outperforms decomposition method as well as earlier methods significantly. Moreover, possible saving of terms can be made more noticeable as we increase the prime factor.

Key–Words: decomposition method, elliptic curve cryptography, non-adjacent form, binary method, NP-hard

1 Introduction

An addition chain is defined as a sequence of integers $a_0, a_1, \ldots, a_r$ starting from $1 = a_0$ and ending with $n = a_r$ with only addition and doubling operations of two previous terms are permitted. This topic has emerged for more than 100 years ago [6] and the field has gradually established itself after it has found its applicability in elliptic curve cryptography (ECC) [1, 2]. In ECC, addition chain method is used to compute the most important operation namely scalar multiplication denoted as $Q = nP$ where P and Q are points on elliptic curve and n is a very large positive integer. Direct multiplication consumes tremendous amount of computational resources, whereas, naive method that performs the addition of P for $n-1$ times is considered inefficient especially when n gets large. The best solution is to use addition chain method.

Although the hardness of finding an optimal chain for any integer n was not proven to be NP-complete, [3] has proved for the generalized case, that finding optimal addition chains for each integer in a sequence, which they regarded as addition sequence problem, is NP-complete. As a result, many heuristic methods such as binary method [4], non-adjacent form method [8, 16, 15, 12, 7, 14], mutual-opposite form method [15, 10] and complementary recoding method [11] were introduced and each method works well in some occasions.

Recently, another heuristic method called decomposition method (DM) [5] was introduced. For an integer n, DM works out an addition chain from its prime factors. Moreover, it takes an input in the form of rules which is unique to each prime, different from earlier methods which were based on binary representation. DM was shown to perform better than previous methods under certain circumstances. However, its capability is likely to be comparable to BM in most cases. Unlike previous methods, DM splits up the problem of generating an addition chain for n into a two-layered approach, the lower layer concerns with individual prime and the upper layer deals with prime powers that builds up n.

Noteworthy, methods we listed above are based on binary representation which uses $\{0, \pm 1\}$ to represent n. There are other representations such as m-ary and also multiple basis, but that are considered to fall under different family. Since computer representation for data is in bits, binary representation has a tendency to be more efficient during preprocessing.

The method to be proposed namely signed decomposition method (SDM), based on binary representation, is an improvement to the DM. Initially, a list of primes $p_1, p_2, \ldots, p_s$ is randomly generated in a form of rules. A prime is uniquely determined by a rule. From each rule, we can generate one specific chain. The size of each prime is proportional to the length of its respective rule and is controllable and can be made suitable for ECC scalar. Let $n = p_1.p_2.\ldots.p_s$, a chain for n can be computed by operating a rule after the other which can be seen as concatenating a chain to another that will produce an ascending sequence of integers similar to that of an addition chain.

This way, SDM does not randomly generate n, form an appropriate representation of which nP can be computed as a sequence of integer point $P, 2P, \ldots, nP$ in similar fashion as that of previous methods. Instead, it randomly generates a list of rules, one for each prime and from these rules, nP can be computed where n itself is the result of multiplying primes altogether.

SDM aims at shortening an addition chain by optimizing the add rule at prime layer by adapting the idea of non-adjacent form [9]. This method can also be seen a hybrid of decomposition method and non-adjacent form method. Some analysis will be performed to determine the boundary value of this method. For the purpose of performance comparison, SDM is tested against other methods that fall within the same family.

From security perspective, if we use this method for message encryption as that found in ECC ElGamal analogue, as an example, we affirm that this method does not introduce any new loopholes. The fact that rules are not interchanged between parties, rather be wrapped up into an encrypted message will maintain the security of ECC.

The remaining of this paper is organised as follows. Section 2 starts off by revisiting DM. Section 3 introduces the idea behind SDM and its translation into some mathematical formulation. Section 4 develops the whole idea of SDM into a computational model. Section 5 analyzes the boundary of an addition subtraction chain produced by SDM. Section 6 includes a comparative results on SDM against other earlier methods. Finally, Section 7 serves as a summary to the findings.

2 Decomposition Method

A decomposition method (DM) was introduced as an alternative to other existing methods. Instead of generating n, this method begins with a list of primes and its powers that make up n. Each prime is transformed into a rule of which an addition chain for n can be generated from the combination of these rules. However, the work of converting a prime to a rule requires some precomputations. It was also mentioned about generating rule or a combination of rules as a replacement to different p that make up n. DM uses these rules to generate an addition chain for n. By doing this, the efficiency will be improved as the needs for precomputation can be avoided. Furthermore, the idea adds up an extra security to the system, where unlike number, rule cannot easily be understood and manipulated by adversaries. In this paper, a study of DM will be advanced further, and this starts with a redefinition of rule from DM. Prior doing that, some necessary conditions must first be determined.

A rule will normally be generated at random by some random generator. It constitutes of dbl and add elements representing doubling and addition operations respectively. Each rule starts with a list of dbl's followed by a number of add's. Doubling always involves the immediate predecessor term. The number of dbl's denoted as $\#dbl$ specifies the range for p in that $2^{\#dbl} \leq p \leq 2^{\#dbl+1}$. The following number of add's denoted as $\#add$ cannot be more than the number of dbl's such that $0 \leq \#add < \#dbl$. Within add rule, the first operand always be the immediate predecessor term while the second operand is taken in descending order from terms generated by dbl rule as the add rule ascends to the right.

The fact that [5] studied rules for prime numbers, similarly in this paper, a study is focused on prime rule. Nonetheless, a generic term 'rule' will be subcategorized into unsigned rule and signed rule which refers to DM and SDM respectively. Definition 1 suggests a more accurate definition for unsigned rule.

Definition 1 *Let p be a prime. An unsigned rule for p denoted by $rule(p)$ is defined as a sequence of dbl's followed by add's of the form*

$$rule(p) = dbl(a_0), dbl(a_1), \ldots, dbl(a_{i-1}), add(a_i, a_{j_1}), add(a_{i+1}, a_{j_2}), \ldots, add(a_{r-1}, a_{j_m})$$

where:
(1) $a_0, a_1 = dbl(a_0), a_2 = dbl(a_1), \ldots, a_r = add(a_{r-1}, a_{j_m})$ is the respective addition chain for which $0 \leq j_m < \ldots < j_2 < j_1 \leq i-1$,
(2) $a_{j_c} > 0$ for all c such that $1 \leq c \leq m$.

Definition 1 describes the complete form of a rule. The combination by means of keeping some terms of $add(a_i, a_j)$ and throwing others makes up a unique p between $2^{\#dbl}$ to $2^{\#dbl+1}$ inclusive.

It has been demonstrated earlier that for some primes, DM produces the shortest chain among others, however in other cases it is not. This is due to the fact that a rule for a prime p does not always produce the shortest chain especially when comparing to NAF. In spite of this, the original idea of DM has opened up an opportunity for study on addition chain problem and an improvement for this shortcoming in DM will be the main topic to be addressed in the remaining sections.

3 Signed Decomposition Method

Signed decomposition method (SDM) is brought forward to address the weaknesses found in DM. SDM accepts an unsigned input rule from DM and in return produces a signed rule. SDM aims at optimizing the add rules within unsigned rule. Obviously, it does not touch the *dbl* rule as there is nothing much that can be done. Similar to the idea of NAF, SDM allows not only addition of terms but also subtracting them. This is valid because subtraction requires no significant extra resource than addition on elliptic curve. By allowing subtraction operation into the picture, an original addition chain can be redefined into an addition subtraction chain. The following study is strictly dedicated for SDM.

Definition 2 *An addition subtraction chain for n is a sequence of positive integers of the form $a_0 = 1, a_1, \ldots, a_r = n$ such that $a_i = a_{i-1} \pm a_j$ where $j \leq i-1 < i$, $0 \leq i \leq r$. The generation of a_i must satisfy the following conditions:*
(1) if $a_{i-1} = a_j$, then $a_i \leq a_r$ or $a_i > a_r$.
(2) if $a_{i-1} \neq a_j$ and $a_i = a_{i-1} + a_j$, then $\neg\exists k > j$ such that $a_i = a_{i-1} + a_k \leq a_r$ or $\neg\exists k < j$ such that $a_i = a_{i-1} + a_k \geq a_r$.
(3) if $a_{i-1} \neq a_j$ and $a_i = a_{i-1} - a_j$, then $\neg\exists k > j$ such that $a_i = a_{i-1} - a_k \geq a_r$ or $\neg\exists k < j$ such that $a_i = a_{i-1} - a_k \leq a_r$.

The following definitions for doubling and addition operations are similar to that found in [5]. We rewrite them here for the purpose of completeness.

Definition 3 *A doubling operation (dbl) for any term a_i in a sequence $a_0, a_1, \ldots, a_r$ is defined as $a_i = dbl(a_{i-1}) = 2a_{i-1}$.*

A doubling operation (*dbl*) for any term a_i is still defined as $a_i = 2a_{i-1}$ as in Definition 2(1), although this time a_i can be greater than a_r. Whereas, an addition operation (add) is still $a_i = a_{i-1} + a_j$ although the decision on a_j must follow Definition 2(2).

Definition 4 *An addition operation (add) for any term a_i is defined as $a_i = a_{i-1} + a_j$, where $0 \leq j < i-1$.*

An original add rule can be optimized into add or sub rules. In addition, following to *dbl* and add rules from DM, a new definition for sub rule is proposed here. Similarly, the choices of a_j within sub rule is decided according to Definition 2(3).

Definition 5 *A subtraction operation (sub) for any term a_i is defined as $a_i = a_{i-1} - a_j$, such that $a_{i-1} - a_j \geq a_r$ where $0 \leq j < i-1$.*

Although the selection on which operation to be operated (*dbl* or (add or sub)) and the element within the add or sub term is now different, they must abide by the conditions given by Definition 2. This selection process ensures the minimality of the addition subtraction chain.

First, let's present an idea behind SDM with some concrete examples. Consider a prime p. During the computation of its addition subtraction chain, to find the next term a_{i+1}, the concept of 'closest' value to p is used. This is a two-step approach where at first a decision on the type of operation for a_{i+1}, *dbl* or add/sub must be made, and if add/sub is chosen, an appropriate second operand corresponds to the operation must follow. To illustrate this idea, consider the following examples. First, let $p = 23$. SDM generates the chain of $1, 2, 4, 8, 16, 24, 23$. The first four operations were chosen to be *dbl*'s as their values always produce the term closer to 23 than any add/sub rule. At $a_i = 16$, to calculate a_{i+1}, there is a choice to add 4 or 8 to a_i such that a_{i+1} could either be $a_{i+1} = 20 < p$ or $a_{i+1} = 24 > p$. Here 8 is chosen because 24 is closer to 23 than to 20. For the second example, let $p = 31$. This method produces the chain of $1, 2, 4, 8, 16, 32, 31$. Again, the first four terms were generated as a result of *dbl* operations. At $a_i = 16$, to find a_{i+1}, there is a choice to add 8 or 16 to a_i such that $a_{i+1} = 24 < p$ or $a_{i+1} = 32 > p$. In this case, 16 is chosen because 32 is closer to 31 than to 24, and the resulting operation is instead a doubling.

Definition 6 *Let p be a prime. A term $a_{i+1} = a_i \pm a_j$ for which $0 < j \leq i$ is considered as the closest value to p if $|p - (a_i \pm a_j)| < |p - (a_i \pm a_k)|$ for any $k \neq j$.*

Remark 7 *If p is equidistant from both sides, the lower value is considered as the 'closest value'.*

Meanwhile, to show SDM surpasses others, consider $p = 47$. This method produces a chain 1, 2, 4, 8, 16, 32, 48, 47 of length 7 whereas BM, NAF and CR produce chains of length 9, 8 and 8 respectively.

It was shown earlier that, to calculate the next term based on add or sub rule, two operands will be combined together by means of addition or subtraction respectively. The value of p will be in between $add(a_i, a_{j_r}) < p < add(a_i, a_{j_{r+1}})$ where a_i is the immediate predecessor and a_{j_r} and $a_{j_{r+1}}$ are those terms generated as a result of doubling operations earlier. The decision to select a_{j_r} or $a_{j_{r+1}}$ will be determined by their closeness to p for which the closer value is to be chosen. However, there are cases when p is equidistant from both sides. For the purpose of implementation, the lower value is chosen. In a case where $a_i > p$, the coming operation for a_{i+1} must be a sub, otherwise it will be an add. Note that, there is also a case where a presumed add becomes a dbl due to adding a_i to itself as shown in the second example.

Let's generalized p to n for a moment. Consider addition rules for small n. Table 1 provides an individual chain for $2 \leq n \leq 16$ which separates dbl and add/sub sequence into different columns. Under Column 1 comes the dbl rule, which is similar for both DM and SDM, under Column 3 are add rules for DM and under Column 4 are add/sub rules belongs to SDM.

Observe closely that the pattern for add rules for the first 2^{i-1} elements based on $dbl = 2^i$ can be seen to be elements based on $dbl = 2^{i-1}$. As an example, add rules for $n = 9, 10, 11, 12$ based on $dbl = 1, 2, 4, 8$ have a pattern of add rules for $n = 5, 6, 7, 8$ respectively, based on $dbl = 1, 2, 4$. Studies undertaken on Column 3 and Column 4 have discovered two interesting findings. The first one says that, if two consecutive add rules in Column 3 uses two consecutive doubling terms as their second operands, it can be exchanged without any losses to the add/sub rules shown in Column 4. The idea applies to $n = 7, 11, 14$. Another one says that, if at least three consecutive add rules in Column 3 uses three consecutive doubling terms as their second operands, it can be exchanged to the add/sub rules in Column 4 for some gains. This idea applies to $n = 15$. From these observations, it can be concluded that for the first case, one need not do anything unless by so doing will trigger another substitution which will optimize the rule. But for the second case, one should transform the rule on Column 3 into the rule on Column 4. As a result of the first case, one can avoid the disadvantage of NAF which unnecessarily substitutes two consecutive nonzero terms. As an example, NAF substitutes 10011 with $1010\bar{1}$ which produces no gain. Moreover, having n in a form of rule eliminates the possibility of adding an extra digit to the left of most significant bit in a case where two or more consecutive 1's appear as that of NAF. As an example NAF substitutes 1100 with $10\bar{1}00$. For the simplicity of SDM implementation, all block of two 1's will be substituted even though it does not induce another substitution. This does not affect the original idea of 'closest' value presented earlier as both techniques generate the same number of terms.

Based on the discussion above, a proposed iterative method to generate a signed rule from an unsigned rule of DM is presented here. Considering a rule for p as $rule(p)$, such an iterative process can be described as follows.

Let

$rule(p) = dbl(a_0),\ dbl(a_1),\ \ldots,\ dbl(a_{i-1}),\ add(a_i, a_{j_1}), add(a_{i+1}, a_{j_2}), \ldots, add(a_{r-1}, a_{j_m})$ where $0 \leq j_m < \ldots < j_2 < j_1 \leq i-1$.

Step 1: The process starts at rule $add(a_{r-1}, a_{j_m})$ and move one rule to the left each time until rule $add(a_i, a_{j_1})$. For c from m to 1, if found $a_{j_c} \neq 0$, scan $a_{j_{c-1}}, a_{j_{c-2}}$ and so on until zero digit is found.

Step 2: If $a_{j_c}.a_{j_{c-1}}.\ldots.a_{j_{c-l}} \neq 0$ with $l+1$ consecutive terms such that $l+1 \geq 2$. Let $i < t < r-1$, then apply the following substitutions:

$a_{j_{c-l}} \Leftarrow a_{j_{c-(l+1)}}$ such that $add(a_t, a_{j_{c-l}})$ becomes $add(a_t, a_{j_{c-l}+1})$.

$a_{j_c} \Leftarrow -a_{j_c}$ such that $add(a_{t+l}, a_{j_c})$ becomes $sub(a_{t+l}, \overline{a_{j_c}})$.

$\{a_{j_{c-l}}, a_{j_{c-(l-1)}}, a_{j_{c-(l-2)}}, \ldots, a_{j_{c-1}}\} \Leftarrow 0$ such that all terms $add(a_{t+1}, a_{j_{c-l}})$, $add(a_{t+2}, a_{j_{c-(l+1)}})$, $\ldots, add(a_{t+(l-1)}, a_{j_{c-1}})$ are eliminated.

Step 3: The first operand for rules starting from $add/sub(a_{t+l}, a_{j_c})$ through to the right most must be shifted to the left $l-1$ steps such that $add/sub(a_{t+l}, a_{j_c})$ becomes $add/sub(a_{t-1}, a_{j_c})$ and so on. This value represents the number of add rules that was cancelled during substitution.

Step 4: Repeat Steps 1-3 until no more 3 consecutive terms of the second operand exists.

Another example, consider an integer $p = 239_{10} = 11101111_2$. The unsigned rule for 239 generated by DM is

$rule(239) = dbl(a_0),\ dbl(a_1), dbl(a_2),\ dbl(a_3),\ dbl(a_4),\ dbl(a_5),\ dbl(a_6),\ add(a_7, a_6),\ add(a_8, a_5), add(a_9, a_3), add(a_{10}, a_2), add(a_{11}, a_1),\ add(a_{12}, a_0)$.

Applying the algorithm for the first substitution yields

$rule(239) = dbl(a_0), dbl(a_1), dbl(a_2), dbl(a_3),\ dbl(a_4), dbl(a_5), dbl(a_6), add(a_7, a_6), add(a_8, a_5),\ add(a_9, a_4), sub(a_{10}, a_0)$.

Applying the algorithm for the second substitution yields

$rule(239) = dbl(a_0), dbl(a_1), dbl(a_2), dbl(a_3),\ dbl(a_4), dbl(a_5), dbl(a_6), add(a_7, a_7), sub(a_8, a_4),\ sub(a_9, a_0)$.

At this end, no more sub is possible. The result of the algorithm above is regarded as the signed rule of

Table 1: Addition rule for $n \leq 16$

dbl rule	n	DM add rule	SDM add/sub rule
$dbl(a_0)$	3	$add(a_i, a_0)$	
	4	$add(a_i, a_1)$	
$dbl(a_0), dbl(a_1)$	5	$add(a_i, a_0)$	
	6	$add(a_i, a_1)$	
	7	$add(a_i, a_1)$+$add(a_i, a_0)$	$add(a_i, a_2)$+$sub(a_i, a_0)$
	8	$add(a_i, a_2)$	
$dbl(a_0), dbl(a_1), dbl(a_2)$	9	$add(a_i, a_0)$	
	10	$add(a_i, a_1)$	
	11	$add(a_i, a_1)$+$add(a_i, a_0)$	$add(a_i, a_2)$+$sub(a_i, a_0)$
	12	$add(a_i, a_2)$	
	13	$add(a_i, a_2)$+$add(a_i, a_0)$	
	14	$add(a_i, a_2)$+$add(a_i, a_1)$	$dbl(a_i, a_3)$+$sub(a_i, a_1)$
	15	$add(a_i, a_2)$+$add(a_i, a_1)$+$add(a_i, a_0)$	$dbl(a_i, a_3)$+$sub(a_i, a_0)$
	16	$add(a_i, a_3)$	

SDM. As a consequence, an original rule with 13 operations is now consisting of 10 operations, less three operations. From the studies above, a new definition for a signed rule can be suggested.

Definition 8 *Let p be a prime. A signed rule for p denoted by $rule(p)$ is defined as a sequence of dbl's followed by add/sub's of the form*

$$rule(p) = dbl(a_0), dbl(a_1), \ldots, dbl(a_{i-1}), add/sub(a_i, a_{j_1}), add/sub(a_{i+1}, a_{j_2}), \ldots, add/sub(a_{r-1}, a_{j_m})$$

where:
(1) $a_0, a_1 = dbl(a_0), a_2 = dbl(a_1), \ldots, a_r = add/sub(a_{r-1}, a_{j_m})$ is the respective addition subtraction chain for which $0 \leq j_m < \ldots < j_2 < j_1 \leq i$,
(2) $a_{j_k} > 0$ for all k such that $1 \leq k \leq m$.

Be it signed or unsigned, similar notation will be used for a rule for p which is $rule(p)$. What really decides the different is the content within the rule. The question arises whether signed rule preserves the uniqueness property as that of unsigned rule. The following theorem have it all.

Theorem 9 *An addition subtraction chain $a_0, a_1, \ldots, a_r$ for a prime p can be computed from a given signed rule and each rule is unique to each p.*

Proof: First, the proof of each signed rule is an addition subtraction chain is laid out. For each value of a_i such that $2a_i < a_r$ and $2a_{i+1} > a_r$, by Definition 2, the maximum number of dbl operations is $i+1$. Similarly, a_0 is always set to 1 and by Definition 8, the first operation is always dbl. As a consequence, $a_0 = 1$ and $a_1 = 2$. Therefore, there always exists a path to $a_r = p$ starting from a_{i+1} or a_{i+2} as a result of addition and subtraction operations.

To prove the uniqueness of this relation, suppose there are two different signed rules for p, both having i and j number of doubling operations. It will be shown that they both are the same.
$rule_1(p) = dbl(a_0), dbl(a_1), \ldots, dbl(a_{i-1}), add/sub(a_i, a_{s_1}), add/sub(a_{i+1}, a_{s_2}), \ldots, add/sub(a_{m_1-1}, a_{s_x})$
$rule_2(p) = dbl(b_0), dbl(b_1), \ldots, dbl(b_{j-1}), add/sub(b_j, b_{t_1}), add/sub(b_{j+1}, b_{t_2}), \ldots, add/sub(b_{m_2-1}, b_{t_y})$
Consider the corresponding chains for both rules
$chain_1(p) = a_0, a_1, a_2, \ldots, a_i, \ldots, a_{m_1} = add/sub(a_{m_1-1}, a_{s_x})$
$chain_2(p) = b_0, b_1, b_2, \ldots, b_j, \ldots, b_{m_2} = add/sub(b_{m_2-1}, b_{t_y})$
If the two chains can be proved to be equal, it can be deduced that both rules are also equal because rule and chain are interchangeable. Let's show that each a_i is a b_j for some i and j from $i = 0, 1, \ldots, m_1$ and $j = 0, 1, \ldots, m_2$. According to Definition 2(1) and condition set by Definition 6, the number of doublings d, for $chain_1(p)$ is equal to that of $chain_2(p)$. Hence, $a_0 = b_0, a_1 = b_1, \ldots, a_d = b_d$ for the first $d+1$ terms where $0 \leq d \leq m_1$. Since the terms a_{s_u} and b_{t_v} for $0 \leq s_u, t_v \leq d$, $1 \leq u \leq x$ and $1 \leq v \leq y$, were a result of previous doublings, by Definition 2(2,3) and condition set by Definition 6, $a_{s_u} = b_{t_v}$. Inductively, from $a_{i+1} = a_i \pm a_{s_x}$ and $b_{i+1} = b_i \pm b_{t_y}$ and since $a_i = b_i$ and $a_{s_u} = b_{t_v}$, this yields $a_{i+1} = b_{i+1}, a_{i+2} = b_{i+2}$, and so forth until

the last term $a_{m_1} = b_{m_2}$.
Let's show that $m_1 = m_2$ by contradiction. Assume $m_1 \neq m_2$ such that $m_1 > m_2$. By the same argument as above, $a_0 = b_0, a_1 = b_1, \ldots, a_{m_2} = b_{m_2}$. For all a_i, $i > m_2$, there exists no b_i such that $a_i = b_i$. This could be the result of different number of doublings or/and the number of additions/subtractions on both chains. By Definitions 6, this will produce two different integers. Hence $m_1 = m_2$. □

Lemma 10 *Suppose a signed rule for a prime p is given by*

$$rule(p) = dbl(a_0), dbl(a_1), \ldots, dbl(a_{i-1}), add(a_i, a_{j_1}), add(a_{i+1}, a_{j_2}), \ldots, add(a_{r-1}, a_{j_m})$$

then the signed decomposition method generates an addition subtraction chain for any composite integer $n = p^e$ in a sequential and $n = a_{er}$.

Proof: By Definition 2, this proof follows similar technique as Lemma 2.11 in [5]. □

Lemma 11 *Suppose signed rules for each primes p_i, for $i = 1, 2, \ldots, s$ gives the following addition subtraction chains*

$$chain(p_i) = a_{i_0}, a_{i_1}, a_{i_2}, \ldots, a_{i_{r_i}}.$$

Then the signed decomposition method generates an addition subtraction chain for $n = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$ in a sequence and $n = a_{\sum_{i=1}^{s} e_i . i_{r_i}}$.

Proof: By Definition 2, Definition 6 and Lemma 10, this proof can be completed in similar technique as Lemma 2.13 in [5]. □

For each signed rule, the average number of addition operation given by this method is $\frac{\#dbl}{3}$, shorter than the average produced by DM which is $\frac{\#dbl}{2}$. In general, it is similar to that of NAF where $\#dbl$ is substituted to l, the length of binary representation except for the case where NAF performs redundant substitution.

4 Algorithm Development

As stated earlier, this study is proposed for decomposed n in prime power $p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$ form. Algorithm 1 takes an input rule$(p_1, p_2, \ldots, p_s)$ and generates the complete chain for n.

Algorithm 1. Generating chain for n

1. INPUT: $rule(p_1, p_2, \ldots, p_s), e_1, e_2, \ldots, e_s$
2. for i from 0 to $s-1$ step-up by 1
3. for k from 0 to $e_i - 1$ step-up by 1
4. for l from 0 to $c_i - 1$ step-up by 1
5. if rule is dbl
6. $a_{ie_ic_i+kc_i+l+1} = 2.a_{ie_ic_i+kc_i+l}$
7. else if rule is add
8. $a_{ie_ic_i+kc_i+l+1} = a_{ie_ic_i+kc_i+l} + a_{ie_ic_i+kc_i+j}$
9. else if rule is sub
10. $a_{ie_ic_i+kc_i+l+1} = a_{ie_ic_i+kc_i+l} - a_{ie_ic_i+kc_i+j}$
11. OUTPUT: $a_0, a_1, \ldots, a_r = n$

We use array to store all the terms generated from the chain starting from a_0. This is necessary as the add rule will use some of the previous terms, if not all in its addition or subtraction operations. There will be one rule assigned for each prime p_i. This rule is executed e_i number of times. For each i, let $c_i = \#(dbl + add)_i$ be the number of doubling and addition operations for p_i. The code seems a bit complicated with 3 nested loops but the complexity is approximated to $c_i.e_i.s$.

5 Analysis

Earlier studies show that the length of an addition chain produced by DM, $l_{dm}(n)$ is bounded by the same boundary as that of optimal addition chain $l(n)$ for which $l_{dm}(n) \geq l(n)$. For a quick recap, consider $n = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$ such that $2^m + 1 \leq n \leq 2^{m+1}$. The boundary for $l_{dm}(n)$ is given by $m + 1 \leq (m_1.e_1 + m_2.e_2 + \ldots + m_s.e_s) + 1 \leq l_{dm}(n) \leq 2(m_1.e_1 + m_2.e_2 + \ldots + m_s.e_s) \leq 2m$.

Thanks to P. Erdös, the study for the length of an addition subtraction chain generated by SDM, denoted as $l_{sdm}(n)$ is made easier. In his paper, [17] proved that the length of optimal addition subtraction chain, denoted as $l'(n)$ is always shorter than, if not equally lengthy to an optimal addition chain $l(n)$ such that $l'(n) \leq l(n)$. Due to this the following theorems will be stated without further proofs.

Lemma 12 *Let p be an odd prime, $l_{sdm}(p) \leq l_{dm}(p)$.*

Proof: Proof is deducible from [17]. □

By Lemma 10, given a signed rule for p of length r, SDM computes the chain for p^e giving the length as e multiple of the length for chain p such that $l_{sdm}(p^e) = er = e \times l_{sdm}(p)$.

Lemma 13 *Let p be an odd prime, $l_{sdm}(p^e) \leq l_{dm}(p^e)$ for $e \in \mathbb{Z}^+$.*

Proof: Since $l_{sdm}(p^e) = el_{sdm}(p)$. By Lemma 12, $el_{sdm}(p) \leq el_{dm}(p)$ which yields $l_{sdm}(p^e) \leq l_{dm}(p^e)$. □

By Lemma 11, given a signed rule for p_i of length r_i for $1 \leq i \leq s$, SDM computes the length of an addition subtraction chain for $n = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$, giving the length as a summation of e_i multiple of the length for chain p_i, for all i such that $l_{sdm}(n) = \sum_{i=1}^{s} e_i i_{r_i} = \sum_{i=1}^{s} e_i l_{sdm}(p_i)$.

Theorem 14 *Let $n = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$. $l_{sdm}(n) \leq l_{dm}(n)$ for $e_1, e_2, \ldots, e_s \in \mathbb{Z}^+$.*

Proof: Earlier we have $l_{sdm}(p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}) = l_{sdm}(p_1^{e_1}) + l_{sdm}(p_2^{e_2}) + \ldots + l_{sdm}(p_s^{e_s}) = e_1 l_{sdm}(p_1) + e_2 l_{sdm}(p_2) + \ldots + e_s l_{sdm}(p_s)$ and by Lemma 12, for all i, $e_i l_{sdm}(p_i) \leq e_i l_{dm}(p_i)$, it is safe to write $e_1 l_{sdm}(p_1) + e_2 l_{sdm}(p_2) + \ldots + e_s l_{sdm}(p_s) \leq e_1 l_{dm}(p_1) + e_2 l_{dm}(p_2) + \ldots + e_s l_{dm}(p_s)$. Grouping back, $l_{sdm}(p_1^{e_1}) + l_{sdm}(p_2^{e_2}) + \ldots + l_{sdm}(p_s^{e_s}) \leq l_{dm}(p_1^{e_1}) + l_{dm}(p_2^{e_2}) + \ldots + l_{dm}(p_s^{e_s})$ will yield $l_{sdm}(p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}) \leq l_{dm}(p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s})$. □

The relationship of the four different chains follows these inequalities $l'(n) \leq l(n)$ and $l_{sdm}(n) \leq l_{dm}(n)$. Although in general, the case of $l_{sdm}(n) \leq l(n)$ (or otherwise) is not always true.

As an extension to the studies on DM, the boundary for the length of an addition subtraction chain generated by SDM can also be determined. For an integer in certain number range, of which studies was initiated by [13], the following assertions can be made.

Lemma 15 *Given an odd prime p, $m + 1 \leq l_{sdm}(p) \leq 2m$ for $2^m + 1 \leq p \leq 2^{m+1}$.*

Proof: Simply substituting $l_{sdm}(p)$ in place of $l_{dm}(p)$ as a consequence of Lemma 12 and Lemma 4.3 in [5] completes the proof. □

Lemma 16 *Let $n = p^e$ such that $2^m + 1 \leq n \leq 2^{m+1}$, if $m_p + 1 \leq l_{sdm}(p) \leq 2m_p$ then $m + 1 \leq em_p + 1 \leq l_{sdm}(p^e) \leq 2em_p \leq 2m$, where m_p is associated with chain generated by SDM.*

Proof: Simply substituting $l_{sdm}(p)$ in place of $l_{dm}(p)$ as a consequence of Lemma 13 and Lemma 4.6 in [5] completes the proof. □

Theorem 17 *Let $n = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$ such that $2^m + 1 \leq n \leq 2^{m+1}$. Then $l_{sdm}(n)$ is bounded as $m + 1 \leq (m_1.e_1 + m_2.e_2 + \ldots + m_s.e_s) + 1 \leq l_{sdm}(n) \leq 2(m_1.e_1 + m_2.e_2 + \ldots + m_s.e_s) \leq 2m$ where $2^{m_i} + 1 \leq p_i \leq 2^{m_i+1}$ for all $1 \leq i \leq s$.*

Proof: Simply substituting $l_{sdm}(p)$ in place of $l_{dm}(p)$ as a consequence of Lemma 14 and Lemma 4.7 in [5] completes the proof. □

Instead of depending solely on the properties of binary representation for n, SDM opens up a new window of improving an addition chain by decomposing n into two layers, an individual prime and prime powers.

6 Results

A series of rigorous test was conducted to compare SDM against previous methods such as BM, NAF, CR and DM, for integers from 2 to 1000000. Such limit for the test was chosen for our convenience, as the use of bigger numbers would require tremendous amount of additional computing time. Initially, we investigate the chains generated by SDM against previous methods. This is followed by a test to simulate real world application for large integer n which are randomly generated.

We conducted a series of tests to examine the properties of an addition subtraction chain produced by SDM against other previous methods. Each of these tests takes an input n in a form of $p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$. The result shows that chains generated by SDM are shorter than the previous methods 421498 times, the previous methods produce shorter chains than SDM for 338478 times. Meanwhile for the other 240023 values, they produce chains of at equal length.

We also conducted another test to extend the result from [5] to include SDM with prime-power inputs. Similarly, the test takes an input of approximately 100 decimal digit integer, equivalent to 300 bits which is well above the number required by the current standard. Various combination of primes were selected at random to make up n. In some cases, SDM produces a chain shorter than previous methods and the difference can be large.

From Table 2, consider the first row when $n = 3^{15}.17^7.49^{22}.73^{13}.97^{11}$. For this n, SDM produces a chain of 411 terms, which is similar to DM but shorter by 27 terms than NAF. In this case, SDM improves NAF by 6.2 percents.

Table 2: Computational result for n of $\sim$100 decimal digits using different set of primes

Range	Prime	BM	NAF	CR	DM	SDM
$2 \leq p < 100$	$3^{15}.17^{7}.49^{22}.73^{13}.97^{11}$	494	438	492	411	411
$2 \leq p < 1000$	$3^{12}.83^{10}.311^{15}.929^{11}$	475	425	472	452	422
$2 \leq p < 10000$	$11^{5}.521^{23}.7177^{10}$	543	469	515	438	438
$2 \leq p < 100000$	$5^{21}.239^{17}.59113^{9}$	491	441	486	500	431
$2 \leq p < 1000000$	$127^{11}.917713^{13}$	505	445	499	457	413
$100 \leq p < 1000$	$101^{10}.353^{5}.709^{8}.929^{15}$	507	460	491	444	444
$1000 \leq p < 10000$	$1553^{9}.5521^{13}.9113^{5}$	479	432	489	438	428
$10000 \leq p < 100000$	$49201^{12}.99989^{8}$	487	422	472	412	412
$100000 \leq p < 1000000$	851419^{15}	455	396	432	495	375

Consider the fifth row when $n = 127^{11}.917713^{13}$, it can be seen that SDM again produces the shortest chain among others, with 413 terms. However, DM produces a chain of 457 terms while NAF of 445. Against DM, we figure that NAF introduces an improvement of 2.6 percents (by 12 terms), while SDM of 9.6 percents (by 44 terms). Moreover, for the same n, SDM also improves NAF by 7.2 percents (by 32 terms). This is an example where SDM closed up some weaknesses in DM as well as outclassed the performance of NAF.

In our studies, we specifically compare our result with NAF because NAF is considered as the best among other existing methods. Empirically, SDM outperforms previous methods in its ability to generate shorter chains with significant improvement.

7 Conclusion

Subtraction operation is employed within signed decomposition method as an aggregation to the original doubling and addition operations within decomposition method. Initial DM is a two-layered approach. In this studies, SDM aims at improving chain at prime layer. The obtained result agrees with the theoretical findings introduced at the earlier section. In cases studied, SDM is the method that produces shorter addition subtraction chain than older methods. Apart from efficiency, it also provides an extra layer of protection through an unusual form of rule which cannot easily be manipulated as of number.

References:

[1] N. Koblitz. *Elliptic curve cryptosystems*. Mathematics of Computations. 1987. 48(177) : 203-209.

[2] V.S. Miller. *Use of elliptic curves in cryptography*. Proc. Of Crypto'85. 1985. LNCS 218 : 417-426.

[3] P. Downey, B. Leong, and R. Sethi. *Computing sequences with addition chains*. SIAM J. Computing. 1981. 10(3) : 638-646.

[4] D.E. Knuth. *The art of computer programming, Volume 2: Seminumeral Algorithms*. Addison-Wesley, 1981.

[5] M.A. Mohamed, M.R. Md Said, K.A. Mohd Atan, and Z. Ahmad Zulkarnain. *Shorter Addition Chain for Smooth Integers Using Decomposition Method*. Int. J. Comp. Math. 2011. 88(15) : 2222-2232.

[6] H. Dellac. *Question 49*. L'Intermediaire Math. 1894. 1 : 20.

[7] J.A. Solinas. *Efficient arithmetic on Koblitz curves*. Designs, Codes and Cryptography. 2000. 19 : 195-249.

[8] B. King. *w-NAF, an efficient left-to-right signed digit recoding algorithm*. Proc. ACNS. 2008. LNCS 5037 : 429-445.

[9] F. Morain, and J. Olivos. *Speeding up the computations on an elliptic curve using addition-subtraction chains*. Theoretical Informatics and Applications. 1990. 24(6) : 531-544.

[10] K. Okeya. *Signed binary representations revisited*, Proceedings of Crypto' 04. 2004. 123-139.

[11] P. Balasubramaniam, and E. Karthikeyan. *Elliptic curve scalar multiplication algorithm using*

complementary recoding. Applied Mathematics and Computation. 2007. 190 : 51-56.

[12] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. LMS Lecture Note Series 265. Cambridge University Press, 2004.

[13] A.T. Brauer. *On addition chains*. Bulletin American Mathematical Society. 1939. 45 : 736-739.

[14] M. Joye, and S.M. Yen. *Optimal left-to-right binary signed-digit recoding*. IEEE Trans. Comp. 2000. 49 : 740-748.

[15] R.M. Avanzi. *A note on the signed sliding window integer recoding and a left-to-right analogue*. Proc. SAC. 2005. LNCS 3357 : 130-143.

[16] J.A. Muir, and D.R. Stinson. *Minimality and other properties of the width-w nonadjacent form*. Math. Comp. 2006. 75 : 369-384.

[17] P. Erdos, *Remarks on Number Theory III: On addition chains*. Acta Arith. 1960. 6 : 77-81.