Elliptic Curve Point Multiplication Algorithm Using Precomputation

HANI MIMI, AZMAN SAMSUDIN, SHAHRAM JAHANI
School of Computer Sciences
UniversitiSains Malaysia
Penang, 11800
Malaysia
hani_mimi@yahoo.com

Abstract: - Window-based elliptic curve multiplication algorithms are more attractive than non-window techniques if precomputation is allowed. Reducing the complexity of elliptic curve point multiplication of the form , which is the dominant operation in elliptic curve cryptography schemes, will reduces the overall complexity of the cryptographic protocol. The wBD is a new window-based elliptic curve multiplication method. It is based on new recoding method called window big-digit (wBD). The wBD is a bidirectional method that can be calculated in both directions based on the amount of the available memory.The available memory is invested in an efficient way since wBD has little number of precomputed points compared to other window methods wich make it more suitable for limited storage devices.The BD recoding method requires only one pass to transform the exponent *k* from its binary representation to its wBD representation. Moreover, the wBDkeys has the lowest zero-run length among other window methods. Finally, the number of elliptic curve operations in addition to the execution timeof wBD method is measured. Consequently, the wBD is efficient as other window-based methods.

Key-Words: -Window methods, Single Scalar EC multiplication, Big-digit Recoding, Public key cryptography

1 Introduction

Elliptic curve operations may be improved by many techniques[1]. Finding a new recoding method which transforms the exponent k to k' with less hamming weight is one of the techniques that may crucially affect the efficiency of an EC scheme. These recoding methods are classified into window and non-window methods. Window methods are considered a generalization of non-window methods [2-7].Window methods are used if it is allowed to store some precomputed values [8]. It is also of interest to have a left-to-right recoding method since it enhances the efficiency of computing kP due to the fact that no need to store the recoded exponent k. Figure 1, shows two recoding methods of the integer k; unsigned binary representation$k_2 = \sum_0^{n-1} k_i 2^i , k_i \in \{0,1\}$,and unsigned window representation$k_w = \sum_0^{m-1} k_i 2^{w*i}, k_i \in D_w = \{0,1,\dots,2^w - 1\}$. The hamming weight is $W(k_w) = \frac{n}{w}$. The number of EC doublings relies on the length of the exponent, while the number of EC additions relies on $W(k)$. Hence, processing w digits at a time will reduce the number of EC additions with extra memory needed to store the set D_w. If only odd values of set D_w are used and zero runs are skipped, then the number of additions will be reduced. Moreover, if signed values are also used, the number of precomputationscan be reduced. The exponent k can be scanned left-to-right or right-to-left. The former method is preferable for window EC multiplication methods since it can be combined with EC multiplication methods without storing the exponent k, i.e. left-to-right methods enable us to do recoding and multiplication simultaneously.

k'_{m-1}				…	k'_1				k'_0			
k_{n-1}	…	k_1	k_0	…					k_{w-1}	…	k_1	k_0

Figure 1Binary and Window Representation of an Integer key (k)

Generally, there are two ways for applying window methods: the first one is fixed window method such as the m-ary method, which process w digits at a time without skipping any digit. The second one is applying a more dynamic technique over the recoded exponent which is sliding window method. [5]. Zero runs are skipped while applying the second window technique, therefore only odd window values will be precalculated and stored [9]. Moreover, using singed binary representation will reduce the number of precomputed elements.

Since we are concerned in software implementation issues, the EC defined over prime fields is of our interest. Therefore, the elliptic curve parameters

concerns curves defined over prime fields. A Weierstrass equation is simplified in order to facilitate the usage of elliptic curve equation in elliptic curve cryptography. The following equation is defined over the prime field F_p with characteristic >3.

$$E: y^2 mod p = (x^3 + ax + b)\ mod p$$

Where $a, b, x, y \in F_p$ and $\Delta = -16(4a^3 + 27b^2)$

Let $G = (x_1, y_1)$ and $Q = (x_2, y_2)$ where $G \neq -Q$. Then basic EC operations, addition and doubling, are defined as follows:

$R = G + Q = (x_3, y_3)$

$x_3 = (\lambda^2 - x_1 - x_2)\ mod p$

$y_3 = (\lambda(x_1 - x_3) - y_1) mod p.$

$$\lambda = \begin{cases} \left(\frac{y_2 - y_1}{x_2 - x_1}\right) mod p & G \neq Q \\ \left(\frac{3x_1^2 + a}{2y_1}\right) mod p & G = Q \end{cases}$$

In 1990, Morain and Olivos[10] firstly suggested to apply the non-adjacent form (NAF) to construct the addition-subtraction chain for point multiplication [11]. Window method, over non-sparse optimal signed binary representation, was firstly proposed by Koyoma and Tsuruoka[3]. Miyaji et. al. proposed wNAF window method for fixed and random EC point [12]. Sliding window method over NAF and wNAF was also introduced by Solinas[2, 13]. The fractional window method was presented by [6] to use the available memory in more efficient way than the previous methods. Later on, some left-to-right window methods have been proposed by Okeyaet al. [5], Avanzi[14], Muir and Stinson [15], and by Khabbazian et al. [16]. Some properties of the proposed methods were proved in the previous.On the other hand, the minimality property of fractional window method was proved by Moller [17]. Some properties of non-sparse optimal singed binary representation and its window method were analyzed by Kong and Li [11]. Muir and Stinson showed that wNAF has a minimal number of nonzero digits [18].

Window-based methods require memory to store the precomputed points (windows). These methods can be classified according to two criteria: flexibility of memory usage and direction. Left-to-right recoding is preferred since it can be merged with the EC multiplication method. It is also considered memory efficient method since it requires only *w* digits to be known when applying the multiplication technique. The methods: m-ary, wMOF , MU, KH, Avanzi, and extended FW are left-to-right methods. While KTNS, wNAF, swNAF, FW, KLNS methods are considered right-to-left. Finally, memory flexible methods are those which can limit their number of precomputed points according to the available memory such as fractional and Khabbazian.

2 Proposed Work

Usually, a radix-2 representation of k is called window representation if $w \geq 2$ and the window values are in the digit set $D_w = \{\mp 1, \mp 3, \dots, \mp 2^w - 1\}$. A new window-based single scalar multiplication method over prime fields and using affine coordinates is proposed. It relies on BD recoding method, which is originallay based on ZOT-binary number system [19].Let $k_2 = \sum_{i=0}^{n-1} a_i 2^i, a_i \in \{0,1\}$ be the binary representation of an integer k. Then $k_w = \sum_{i=0}^{l} b_i 2^i, b_i \in \{0\} \cup D_w, D_w = \{\mp 1, \mp 3, \dots, \mp(2^{w-1} - 1)\}$, is the window NAF representation of *k*.

Algorithm 1: Window Big-Digit Recoding

Input: $k_2 = \sum_{i=0}^{n-1} a_i 2^i, a_i \in \{0,1\}$

Output: $k_{wBD} = \sum_{i=0}^{m-1} \hat{b}_i, \hat{b}_i = (t_i, l_i), t_i \in \{0,1,2\}, l_i \leq w\ for\ t_i \neq 0$

Remark: ϵ denotes empty string

$e \leftarrow 0$

while $(i < n)$

- $i \leftarrow j$
- find the largest j ≤ w such that $O \leftarrow (a_i, \dots, a_{i+j}), a_u = 1, \forall$
- $if\ (O \neq \epsilon)\ k_{wBD}[e].t = 1$
- find the largest j ≤ w such that $T \leftarrow (a_i, \dots, a_{i+j}), a_u * a_{u+1}$
- $if\ (T \neq \epsilon) k_{wBD}[e].t = 2$
- find the largest j such that $Z \leftarrow (a_i, \dots, a_{i+j}), a_u = 0, \forall\ i \leq$
- $if\ (Z \neq \epsilon)$
  - $k_{BD}[e].t \leftarrow 0$
  - $k_{BD}[e].l \leftarrow j + 1$
  - $k_{wBD}[e].l \leftarrow k_{wBD}[e].l + k_{wBD}[e-1].l$
- Increment(e, i)

Return k_{wBD}

Algoirhtm 1 is used to convert a key into its wBD representation $k_{wBD} = \sum_{0 \leq i < m} \hat{b}_i, \hat{b}_i = (t_i, l_i), t_i \in \{0,1,2\}, l_i \leq w$ for $t_i \neq 0, l_i \in N^+$.The length of big-zero is equal to the big-zero digit length in addition to the previous big-digit length. This consideration helps in improving the window method and repeating doublings.

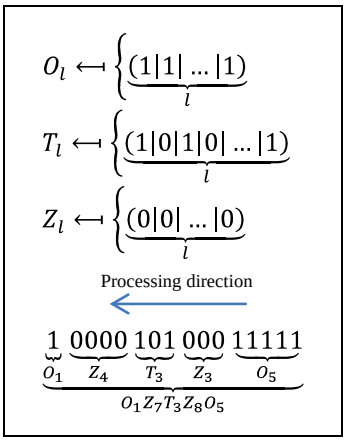


Figure 2 Big-digits Processing and Recognition

The conversion from binary to window BD representation is described in Figure 2. A contiguous sequence of nonzero digits is converted to either big-one or big-two, while the contiguous sequence of zero digits is converted to big-zero. The hamming weight of ZOT-binary number system is $\frac{n}{4.6}$ compared to $\frac{n}{2}$ for binary , $\frac{n}{3}$ for NAF and $\frac{n}{w+1}$ for wNAF [20], where *n* is the bit-length of *k*. The wBD multiplication method is provided in Algorithm 2.

Algorithm 2: Window BD Single Scalar EC Point Multiplication

Input: $G \in E(F_p), w \geq 2, k_{wBD} = \sum_{0 \leq i < n} \hat{b}_i, \hat{b}_i = (t_i, l_i), t_i \in$ *0,1,2, li≤w*
Output: $Q = k_{wBD}G$

Precompute the points:

$$D_w = \{x: x = 2^i - 1, 1 \leq i \leq w\} \cup \{y: y = \frac{1}{3}(2^{i+1} - 1), 3 \leq odd(i) \leq w\}$$

$Q \leftarrow \infty$
For $i(n-1$ *to* $0)$

$$Q \leftarrow \begin{cases} Q + \hat{b}_i(G), & t_i \neq 0 \\ 2^{g_i}(Q), & t_i = 0 \end{cases}$$

Return (Q)

The average length of zero runs in binary, ZOT-binary, NAF, or wNAFrecoded keys is computed using the formula with a key length of $n = \log_2 k$:

$$Z'(k) = \frac{1}{C}\sum_{i=0}^{n-1} 1 - |k_i|\,, \quad C = \sum_{i=1}^{n-1} z(i),$$

$$z(i) = \begin{cases} 1, & b_i \neq 0\,, b_{i-1} = 0 \\ 0, & b_i = 0 \end{cases}$$

.

Whereas average length of zero runs for BD recoded keys is computed using the following formula:

$$Z'(k) = \frac{\sum_{i=0}^{n-1} l_i z(i)}{\sum_{i=0}^{n-1} z(i)}, z(i) = \begin{cases} 1, & t_i = 0 \\ 0, & t_i \neq 0 \end{cases}$$

The set of precomputed windows is $D_w = \{x: x = 2^i - 1, 1 \leq i \leq w\} \cup \{y: y = \frac{1}{3}(2^{i+1} - 1), 3 \leq \text{odd}(i) \leq w\}$. Whereas the number of precomputed points are $\left(w + \left\lfloor \frac{w-1}{2} \right\rfloor\right)$. Since the BD is a bidirectional recoding method, the memory required to store the recoded exponent is *w* whenever left-to-right method used. The cost of wBD multiplication method is computed using the formula $\mathrm{n}[D] + \frac{n}{w(k_{wBD})}[A], n = \log_2 k$. Finally, only one pass is required to transform the exponent *k* from its binary format to its wBD formant.

Table 2 lists window methods and the required memory for storing the exponent *k* in addition to the recoding direction and the number of passes required by an algorithm to convert the exponent *k* to its new format. Bidirectional methods in addition to left-to-right methods require less memory to store the recoded exponent than right-to-left methods. Number of passes means that the conversion requires two sub-conversions for the recoding process. For example if method *A* converts *k* to singed *k*' next it converts singed *k*' to window singed *k*'' then this method requires two passes. As it can be seen from the table, the conversion of the exponent *k* from its binary representation to its wBD representation requires one pass which is considered one of the advantages of wBD method. Moreover, it is a bidirectional method which also does not store the whole recoded key and can be computed in any direction depending on the available amount of memory.

Table 1 Window Methods Summary

Method	Year	Reference	Required memory	Recoding direction	n-Pass
m-ary	1939	[4]	$O(w)$	BI	1
swNAF	NA	[8]	$O(n)$	RTL	1
KTNS	1993	[3]	$O(n)$	RTL	2

wNAF	1997	[13]	$O(n)$	RTL	1
wMOF	2004	[5]	$O(w)$	BI	2
FW	2002	[6], [17]	$O(n)$	RTL	1
KLNS	2005	[11]	$O(n)$	RTL	2
KH	2005	[16]	$O(w)$	LTR	2
MU	2005	[15]	$O(w)$	LTR	1
wBD	2012	[21]	$O(w)$	BI	1

The mathematical representation of k_{wBD} in addition to other window methods is depicted in Table 3.

Table 2 Mathematical Key Representation for Some Window Methods

Method	Representation
m-ary	$k_{mary} = \sum_{i=0}^{l-1} k_i m^i, m = 2^w, k_i < 2^w, l = \left\lceil \frac{\log_2 k}{w} \right\rceil$
swNAF	$k_{swNAF} = \sum_{i=0}^{l-1} k_i 2^i, k_i \text{ is odd}, \lvert k_i \rvert < 2^{w-1}, k_{l-1} \neq 0, \lceil \log_2 k \rceil \leq l \leq \lceil \log_2 k \rceil + 1$
KTNS	$k_{KTNS} = \sum_{i=0}^{l} k_i 2^i, k_i \text{is odd}, \lvert k_i \rvert \leq 2^w - 3, l \leq \lceil \log_2 k \rceil + 1$
wNAF	$k_{wNAF} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert < 2^{w-1}, k_{l-1} \neq 0, l \leq \lceil \log_2 k \rceil + 1$
wMOF	$k_{wMOF} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert < 2^{w-1}, l \leq \lceil \log_2 k \rceil + 1$
FW	$k_{FW} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert \leq 2^w - 3, l \leq \lceil \log_2 k \rceil + 1$
KLNS	$k_{KLNS} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert \leq \frac{5}{6} \cdot 2^w - \frac{1}{3}, l \leq \lceil \log_2 k \rceil + 1$
KH	$k_{KH} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert \leq (2m-1), l \leq \lceil \log_2 k \rceil + 1,$ where $m \leq C$, and C is the maximum number of points that can be stored
MU	$k_{MU} = \sum_{i=0}^{l-1} k_i 2^i, k \text{ is odd}, \lvert k_i \rvert < 2^{w-1}, l \leq \lceil \log_2 k \rceil + 1$
wBD	$k_{wBD} = \sum_{i=0}^{i<m} \hat{b}_i, \hat{b}_i = (t_i, l_i), t_i \in \{0,1,2\}, l_i \leq w \text{ for} t_i \neq 0, l_i \in N^+.$

3 Implementation

Five recommended elliptic curves defined over prime fields published by the national institute of standards and technology (NIST) are used. The NIST recommended curves over prime fields are used in this study [22]. The prime modulus p is Mersenne or Mersenne-like prime [22].

All NIST elliptic curves are defined with $a = -3$ without much loss in generality. This choice yields faster point doublings in Jacobian coordinates

[22]and does not affect the results gained in this study.The value of factor *h* is 1 for NIST fields. The base point used in this study, $P(x, y)$, is given with each NIST recommended EC [23]. The parameters of these elliptic curves can be found in FIPS PUB 186-3[23]. The algorithms used in this research are coded using C++ language, "Microsoft Visual C++ 2008". Besides, it is implemented using MIRACL cryptographic library [24] since it supports EC applications over prime fields. Several experiments, using two different PCs, have been conducted over hundreds of thousands of randomly chosen *n* bit keys. Computers' specifications are summarized inTable 1.

Table 3Computers' Specifications

System Information	Operating System	Processor	Memory GB
PC1	Windows XP Professional SP3	AMD Phenom(tm) 9650 Quad-Core Processor, MMX, 3DNow (4 CPUs), ~2.3GHz	3.6
PC2	Windows XP Professional SP3	Pentium(R) Dual-Core CPU T4200 @ 2.00GHz (2 CPUs)	3

4 Results and Disscussion

Table 4 represents the mathematical representation of the set of precomputed windows for some window methods. The set D_w for wBD method is formulated and shown in the table.

Table 4PrecomputedPoints for Some Window Methods

Method	Precomputed windows set D_w	Number of precomputed points
m-ary	$\{1,2,\dots,2^w-1\}$	2^w-1
swNAF	$\left\{1,2,\dots,\left(\frac{2(2^{\mathrm{w}}-(-1)^{\mathrm{w}})}{3}-1\right)\right\}$	$\frac{1}{3}(2^{\mathrm{w}}-(-1)^{\mathrm{w}})$[8]
KTNS	$\{\pm1,\pm3,\dots,2^w-3\}$	$2^{w-1}-1$ Modified by [11] to be $\frac{5}{6}\cdot 2^{w-1}-1+\frac{(-1)^w}{3}$
wNAF	$\{\pm1,\pm3,\dots,2^{w-1}-1\}$	2^{w-2}
wMOF	$\{\pm1,\pm3,\dots,\pm(2^{w-1}-1)\}$	2^{w-2}
FW	$\{\pm1,\pm3,\dots,\pm(2^{w-1}+m)\}$where m is an odd integer such that $1\le m\le 2^{w-1}-3$ for $w\ge 2$	$2^{w-2}+\frac{m}{2}$ for $w\ge 3$
KLNS	$\left\{\pm1,\pm3,\dots,\pm\frac{5}{6}\cdot 2^w-\frac{1}{3}\right\}$	$\frac{5}{6}\cdot 2^{w-1}-1+\frac{(-1)^w}{3}$
KH	$\{\pm1,\pm3,\dots,\pm(2m-1)\}$	m where $m\le C$, and C is the maximum number of points that can be stored
MU	$\{\pm1,\pm3,\dots,\pm(2^{w-1}-1)\}$	2^{w-2}
wBD	$\{x: x=2^i-1, 1\le i\le w\}\cup\{y: y=\frac{1}{3}(2^{i+1}-1), 3\le odd(i)\le w\}$	$\frac{1}{2}\left(3w-2^{w \bmod 2+(-1)^w}\right)$

The number of precomputed points has been calculated according Table 4 and the results are presented in Figure 3. The figure shows the number of precomputed windows for wBD multiplication method. wBD method has the lowest number of precomputed points over various windows sizes. It slightly increase for larger window sizes. On the other hand the number of windows required by the m-ary window method is considered the maximum. An EC multiplication method with lower number of precomputed points is more suitable for limited memory devices.

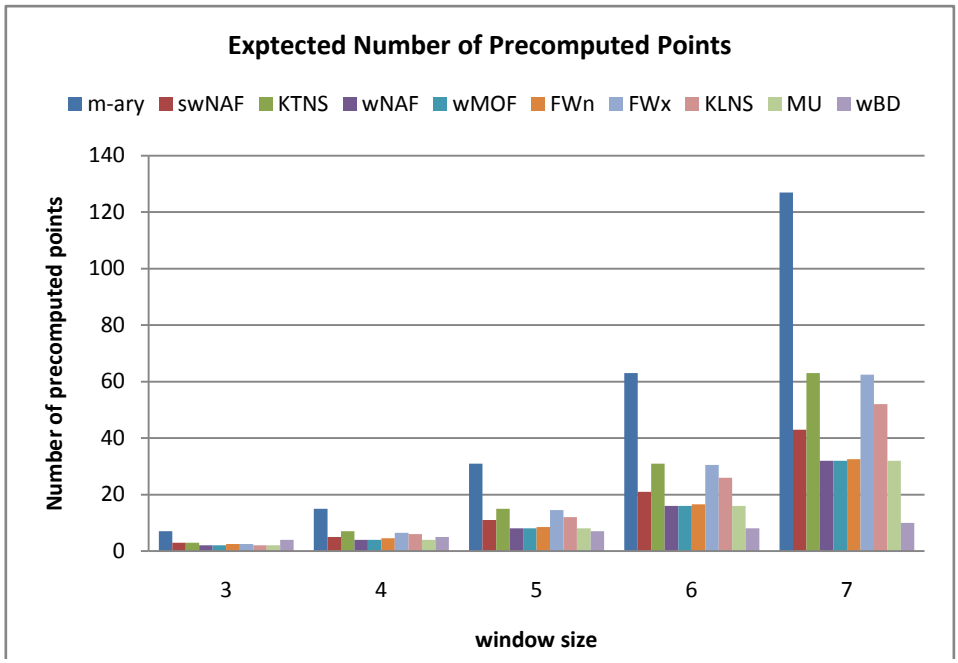


Figure 3 Expected Number of Precomputed Points for Window Methods

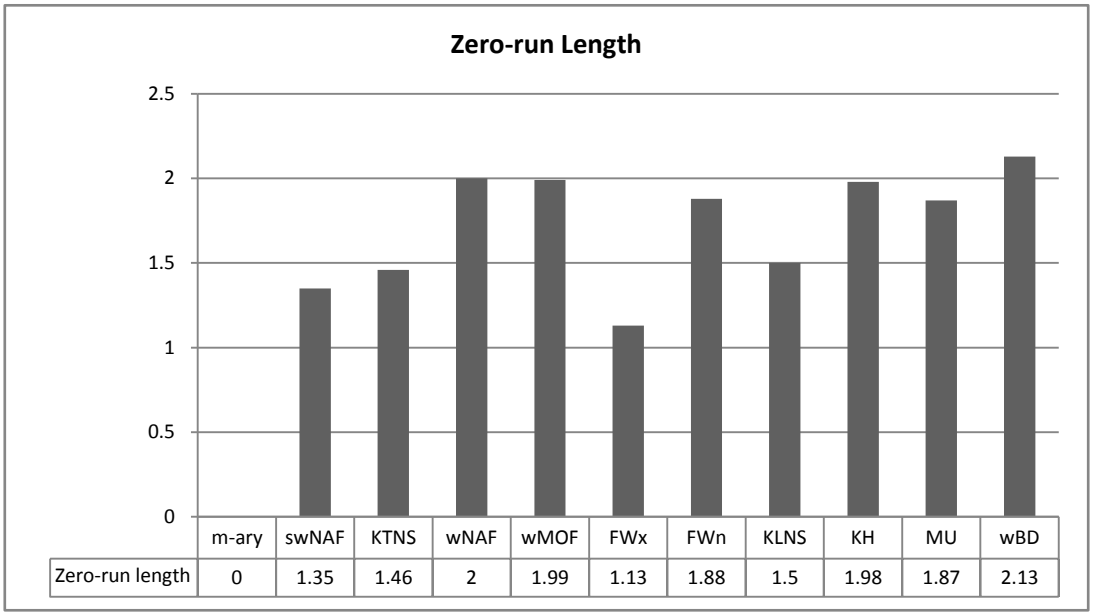
4.1 Zero-run Length

The zero-run length, which is denoted by $Z'(k)$, represents the average length of the zero runs (contiguous sequence of zeros) between windows in the exponent *k*. Increasing zero-run length will speed up the window methods[3]. The expected zero-run length of the binary representation is 1 [11], where it is 1.99 as measured in this study. The expected zero-run length of the NAF representation is 1.35 [11].Table 5 provides a summary of zero-run length for some investigated methods. The table includes mathematical representation and values of this metric. The values of wMOF and MU recoded keys are not determined in the literature. Whereas the formula of wBD method is phrased in the table and the value isexperimentally computed.

Table 5 Length of Zero Runs for Window Recoded Keys

Method	**Zero-run length $Z'(k)$**	**Value for $w = 5$**
m-ary	Zero	0
swNAF	$\frac{4}{3} - \frac{(-1)^w}{3.2^{w-2}}$	1.36
KTNS	1.42	1.42
wNAF	2 [11]	2
wMOF		
FW	$1 + \frac{m+1}{2^{w-1}}$ where m is an odd integer such that $1 \leq m \leq 2^{w-1} - 3$ for $w \geq 2$	1.13 min 1.88 max
KLNS	$1.5\ for\ 3 \leq w \leq 8$[11]	1.5
KH	$\frac{C}{2^{\lfloor \log_2 C \rfloor}} + 1C$ is the maximum number of points that can be stored	2
MU		
wBD	$\frac{\sum_{i=0}^{l-1} l_i z(i)}{\sum_{i=0}^{l-1} z(i)}, z(i) = \begin{cases} 1, & t_i = 0 \\ 0, & t_i \neq 0 \end{cases}$	

The experimental values of zero-run lengths are computed using 100,000 randomly generated keys of size 256 bit and $w = \{4, 5\}$. Figure 4represents the values of this metric that have been calculated. As it can be seen from the figure,wBD method has the maximum length of zero runs. Thisresults is considered an advantage of wBD method since it affects the speed of window method positively as mentioned by Koyoma[3].



	m-ary	swNAF	KTNS	wNAF	wMOF	FWx	FWn	KLNS	KH	MU	wBD
Zero-run length	0	1.35	1.46	2	1.99	1.13	1.88	1.5	1.98	1.87	2.13

Figure 4 Length of Zero Runs in Window Recoded Keys

4.2 Non-zero Density

The non-zero density ($N(k)$) is the ratio of number of non-zero digits to the bit-length of the key, while the hamming weight($W(k)$)is the number of non-zero digits in the recoded exponent. Fewer number of non-zero digits results in fewer number of EC addition required by a multiplication method. Thus, decreasing the value of $W(k)$ or $N(k)$ will increase the efficiency of the EC multiplication method.Most of the gained values are measured theoretically; i.e. by evaluating equations. The non-zero density has two mathematical representations: $N(k) = \frac{1}{w+Z'}$and $N(k) = \frac{W(k)}{n}$. In this section the former representation is used. Therefore, Table 6 shows the inverse of non-zero density ($\frac{1}{N(k)}$).

Table 6 Inverse of Non-zero Density for Some Window Methods

Method	1/Non-zero density
m-ary	w
swNAF	$w + \frac{4}{3} - \frac{(-1)^w}{3.2^{w-2}}$
KTNS	w + 1.42
wNAF	w + 1
wMOF	w + 1
FW	$w + 1 + \frac{m+1}{2^{w-1}}$ where m is an odd integer such that $1 \leq m \leq 2^{w-1} - 3$ for $w \geq 2$
KLNS	$w + \frac{4}{3} + \frac{(-1)^w}{3.2^{w-1}} - \left(\frac{1}{2}\right)^{w-3} + (2 + (-1)^w) \cdot \left(\frac{1}{2}\right)^{\frac{w}{2} - \frac{3}{4}(1-(-1))^w}$

	Or simply: $w + 1.5$
KH	$w + \beta + 1$, where $\beta = \frac{C}{2^{\lfloor \log_2 C \rfloor}} - 1$, and C is the maximum number of points that can be stored
MU	$w + 1$
wBD	$N(k) = \frac{\sum_{i=0}^{n-1} z(i)}{n}, n = \log_2 k,\ z(i) = \begin{cases} 0, & t_i = 0 \\ 1, & t_i \neq 0 \end{cases}$

The inverse of non-zero density is evaluated for $w = 5$ using the equations in Table 6 and the results are shown inFigure 5. It seems that fractional window method with *m* set to maximum value has the lowest value which means that it is expected to have the lowest number of EC additions. On the other hand, if the number of memory locations is limited to 8 locations, i.e. $m = 1$ for FW method, the lowest values are for KLNS and KTNS, while the value of wBD requires experimental computation. Therefore, wBD method in addition to other methods is verified by experimental computations.

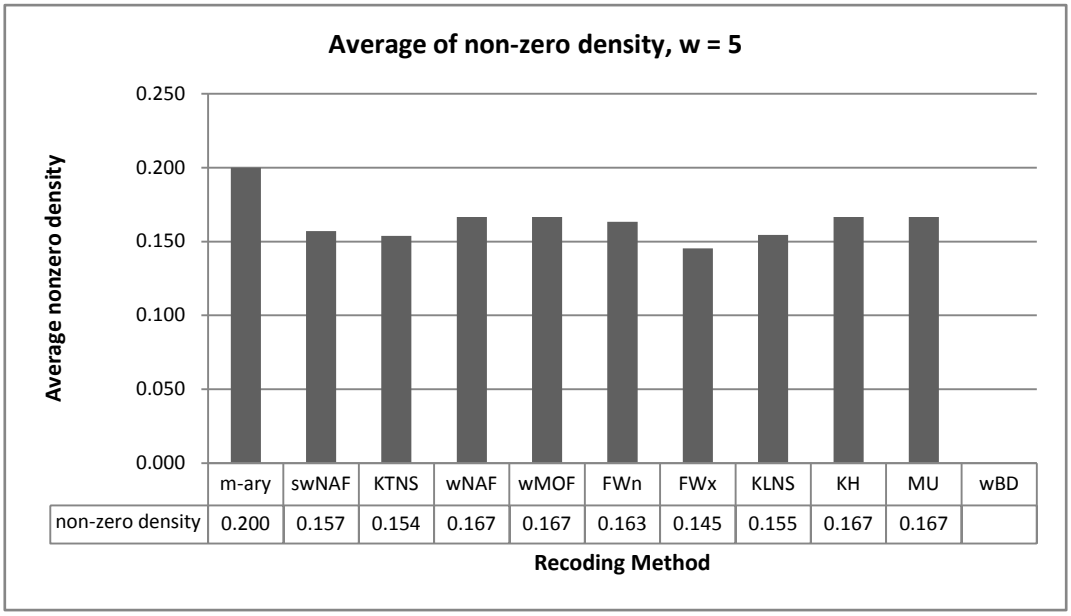


Figure 5 Non-zero Density of Various Recoded Keys – Calculated

An experiment is conducted for measuring the non-zero density for some recoding methods. Each column on the figure represents an average of 10,000 experiments. Randomly generated keys of 224 bit have been examined. The results are presented in Figure 6. Almost the same results are achieved for non-zero density by experimental results.

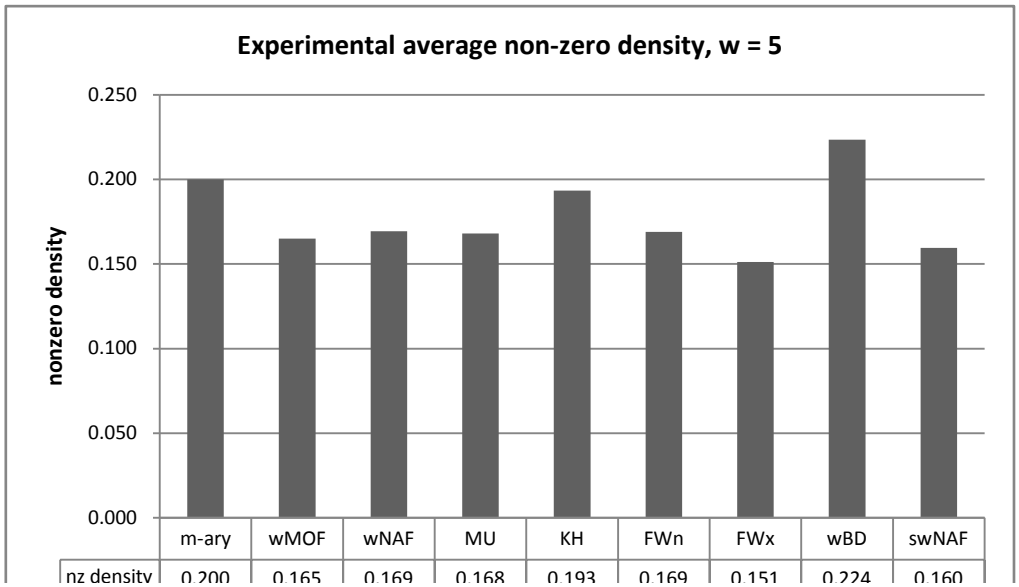


Figure 6Non-zero Density of Various Recoded Keys – Experimental

4.3 Cost of EC Multiplication Methods

The cost of EC multiplication methods are measured in terms of number of EC operations (EC-complexity) and in terms of execution time (time-complexity).The number of EC operations required by each method whenever addition has the same cost as doubling is measured. One cannot compare method A with Method B over different fields. To draw a valid conclusion and to get correct results, both methods should be compared over the same field. Moreover, there is no need to mention the field size when measuring the EC-complexity since it does not affect the results when comparing the methods over the same field.

Table 7 Cost of EC Window Methods - Mathematical Representation

Method	Cost of EC Multiplication
m-ary	$n[D] + d[A], wehre\ d = \left\lfloor \frac{n}{w} \right\rfloor, n = \log_2 k_2$
swNAF	$n[D] + \frac{n}{w + v(w)}[A], where\ \ v(w) = \frac{4}{3} - \frac{(-1)^w}{3.2^{w-2}}$
KTNS	$\left(L - (w - Z')\right)[D] + \left(\frac{L}{w+Z''}\right)[A]$ or simplified as follows $n[D]\ + (\frac{n}{w + Z''})A$
wNAF	$n[D] + \left(\frac{n}{w+1}\right)[A]$
wMOF	$n[D] + \frac{n}{w+1}[A]$
FW	$n[D] + \frac{n}{w + 1 + \frac{m+1}{2^{w-1}}}[A]$
KLNS	$n[D] + \left(\frac{n}{w + \frac{4}{3} + \frac{(-1)^w}{3.2^{w-1}} - \left(\frac{1}{2}\right)^{w-3} + (2 + (-1)^w) \cdot \left(\frac{1}{2}\right)^{\frac{w}{2} - \frac{3}{4}(1-(-1)^w}}\right)[A]$ Or simply $n[D] + \frac{n}{w+1.55}$
KH	$n[D] + \frac{n}{w + \beta + 1}[A], where\ \beta = \frac{C}{2^{\lfloor \log_2 C \rfloor}} - 1$
MU	$n[D] + \left(\frac{n}{w+1}\right)[A]$
wBD	$n[D] + W(k_{wBD})[A]$

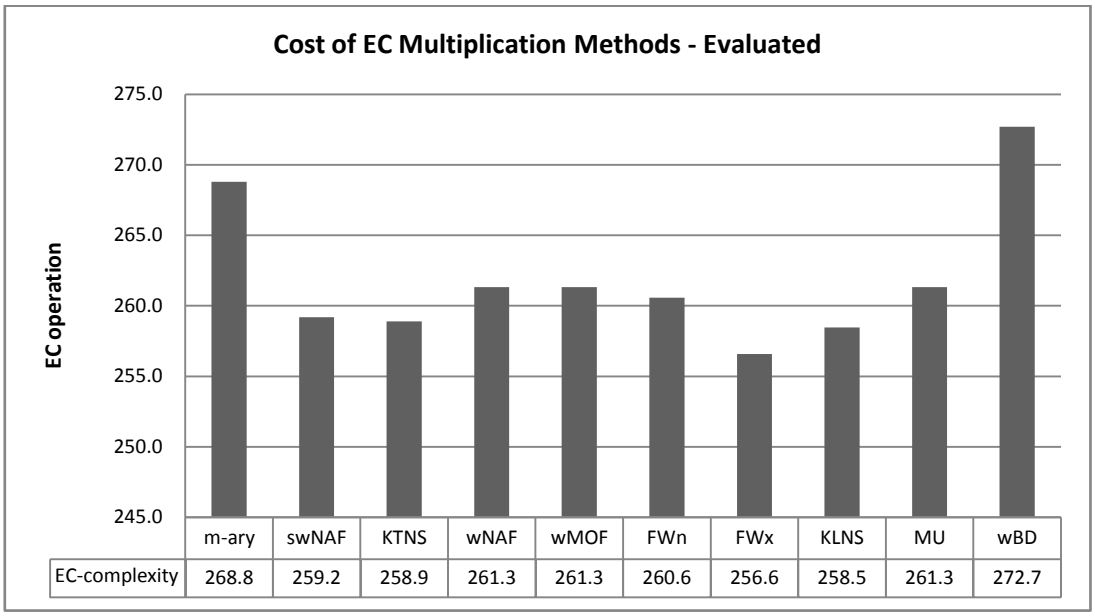


Figure 7 Cost of EC Window Methods in Terms of EC Additions

The number of EC operations required by each method, whenever addition has the same cost as doubling, is shown in Figure 7. These results are calculated using the formulas in Table 7. The window size that is used in this calculation is 5 while the key size is 224 bit. The methods swNAF, KTNS, and KLNS have the lowest cost, while wBD was not the lowest cost method.

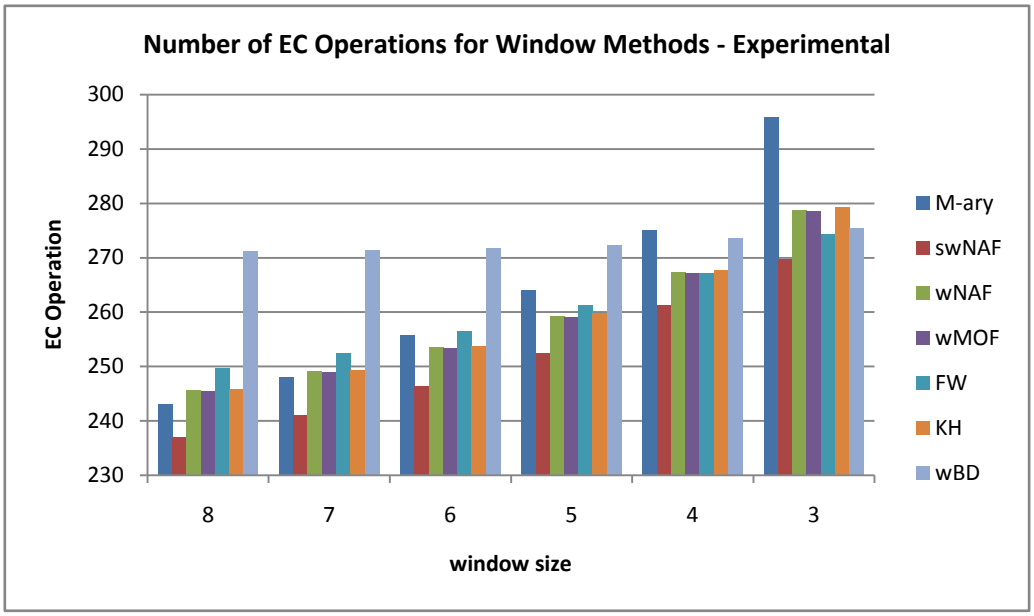


Figure 8 Experimental EC-complexity for Various Window Methods

Figure 8 shows the results of EC-complexity that has been experimentally computed. It represents the number of EC operations required by each method to compute the EC point multiplication *kP*. The experiments are conducted over the NIST prime field of size 521. Each point on the chart represents an average of 10,000 experiments with 224-bit key size.

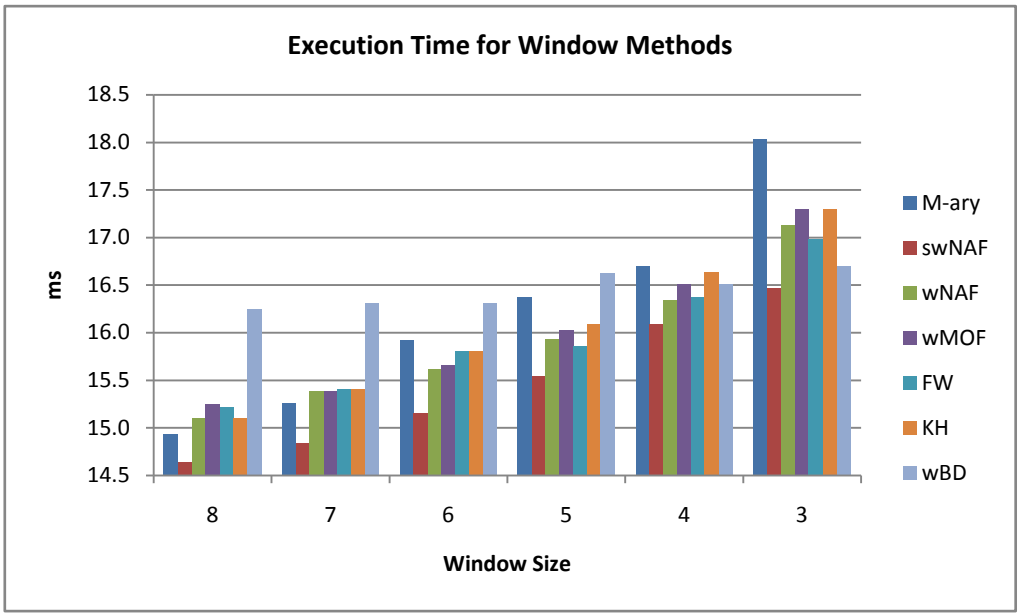


Figure 9 Time-Complexity for Various Window-Based EC Multiplication Methods

The time-complexity is measured and the results are shown in Figure 9. The execution time is measured over various windows sizes ($3 \leq w \leq 8$). The keys are randomly generated and stored in a text file in order not to include its generation time in the total execution time of the whole EC multiplication method. Each point represents the average of 10,000 experiments for each of the evaluated multiplication methods. According to Figure 9, the wBD method is comparable with other EC multiplication methods when $w \leq 4$, while the time-complexity is slightly higher than the other methods for $w \geq 5$.

5 Conclusion

A window-basedbig-digit EC multiplication method is proposed (wBD). One of the advantages of wBD multiplication method is itsbidirectional property. Thus, it is more suitable for memory limited devices. It is a one pass algorithm which means that its recoding process is considered as one the fast recoding methods that requires only one pass to recode the exponent k. The wBD multiplication algorithm is compared to current state of art algorithms based on some defined metrics such as number of precomputed points, EC-complexity and time-complexity.

Regarding precomputations, the precomputed window set is defined. It is found that the number of precomputed windows for wBD EC multiplication method is considered the lowest for various windows sizes. An EC multiplication method with lower number of precomputed points is more suitable for (devices equipped with small memory) limited memory devices.

The zero-run length of wBD keys is defined and the equation is presented in Table 5. The wBDkeys has the lowest zero-run length among other window methods. On the other hand, the nonzero density formula is identified and presented in Table 6. It has been calculated for window methods with $w = 5$. The result of this metric shows that wBD has the highest value which means that it will has the highest number of EC additions. Finally, the EC-complexity and time-complexity costs of elliptic curve methods are measured. Even though wBD multiplication method has a memory advantage against other examined methods, it is comparable with other window methods in terms of EC-complexity and time-complexity. Finally, Further improvement to wBD method may be achieved if more efforts are made to improve the efficiency of composite EC operations.

References

[1] H. Mimi, A. Samsudin, and S. Jahani, "Evaluating Composite EC Operations and their Applicability to the On-the-Fly and Non-Window Multiplication Methods," to appear 2013.

[2] J. A. Solinas, "Efficient Arithmetic on Koblitz Curves," *Designs, Codes and Cryptography,* vol. 19, pp. 195-249, 2000.

[3] K. Koyama and Y. Tsuruoka, "Speeding up Elliptic Cryptosystems by Using a Signed Binary Window Method," *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology,* pp. 345-357, 1993.

[4] I. F. Blake, G. Seroussi, and N. P. Smart, *Elliptic Curves in Cryptography*: Cambridge University Press, 1999.

[5] K. Okeya, K. Schmidt-Samoa, C. Spahn, and T. Takagi, "Signed binary representations revisited," *Proceedings of CRYPTO'04,* vol. 3152, pp. 123-139, 2004.

[6] B. Möller, "Improved techniques for fast exponentiation," *ICISC'02: Proceedings of the 5th international conference on Information security and cryptology,* vol. 2587, pp. 298-312, 2002.

[7] K. Schmidt-Samoa, O. Semay, and T. Takagi, "Analysis of fractional window recoding methods and their application to elliptic curve cryptosystems," *IEEE Transactions on Computers,* vol. 55, pp. 48-57, Jan 2006.

[8] H. Darrel, J. M. Alfred, and V. Scott, *Guide to Elliptic Curve Cryptography*: Springer-Verlag New York, Inc., 2003.

[9] D. M. Gordon, "A survey of fast exponentiation methods," *Journal of Algorithms,* vol. 27, pp. 129-146, Apr 1998.

[10] F. Morain and J. Olivos, "Speeding up the Computations on an Elliptic Curve Using Addition-Subtraction Chains," *Rairo-Informatique Theorique Et Applications-Theoretical Informatics and Applications,* vol. 24, pp. 531-544, 1990.

[11] F. Y. Kong and D. X. Li, "A note on signed binary window algorithm for elliptic curve cryptosystems," *Cryptology and Network Security, Proceedings,* vol. 3810, pp. 223-235, 2005.

[12] A. Miyaji, T. Ono, and H. Cohen, "Efficient elliptic curve exponentiation," *ICICS '97: Proceedings of the First International Conference on Information and Communication Security,* vol. 1334, pp. 282-290, 1997.

[13] J. A. Solinas, "An improved algorithm for arithmetic on a family of elliptic curves," *CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology,* vol. 1294, pp. 357-371, 1997.

[14] R. Avanzi, "A Note on the Signed Sliding Window Integer Recoding and a Left-to-Right Analogue," *Selected Areas in Cryptography,* vol. 3357, pp. 130-143, 2005.

[15] J. A. Muir and D. R. Stinson, "New Minimal Weight Representations for Left-to-Right Window Methods," *CT-RSA'05: Proceedings of the 2005 international conference on Topics in Cryptology,* vol. 3376, pp. 366-383, 2005.

[16] M. Khabbazian, T. A. Gulliver, and V. K. Bhargava, "A new minimal average weight representation for left-to-right point multiplication methods," *Computers, IEEE Transactions on,* vol. 54, pp. 1454-1459, 2005.

[17] B. Möller, "Fractional windows revisited: Improved signed-digit representations for efficient exponentiation," *ICISC'04: Proceedings of the 7th international conference on Information Security and Cryptology,* vol. 3506, pp. 137-153, 2004.

[18] J. A. Muir and D. R. Stinson, "Minimality and other properties of the width-w nonadjacent form," *Mathematics of Computation,* vol. 75, pp. 369-384, 2006.

[19] S. Jahani and A. Samsudin, "ZOT-Binary: A New Numbering System with an Application on Big-Integer Multiplication," *Journal of Theoretical and Applied Information Technology (JATIT),* vol. 48, pp. 029 - 040, 2013.

[20] S. Jahani, "ZOT-MK: A New Algorithm for Big Integer Multiplication," MSc MSc, Department of Computer Science, Universiti Sains Malaysia, Penang, 2009.

[21] H. Mimi, A. Samsudin, and S. Jahani, "Elliptic Curve Point Multiplication Algorithm Using Precomputation," *Security and Communication Networks, to appear*, to appear 2013.

[22] M. Brown, D. Hankerson, J. Lopez, and A. Menezes, "Software implementation of the NIST elliptic curves over prime fields," *Topics in Cryptology*, vol. 2020, pp. 250-265, 2001.

[23] P. Gallagher, D. D. Foreword, and C. F. Director, "FIPS PUB 186-3 Federal Information Processing Standards Publication Digital Signature Standard (DSS)," 2009.

[24] Shamus. (2010, December 16). *MIRACL Library*. Available: http://www.shamus.ie/