International Journal of Technology (2017) 3: 519-527
ISSN 2086-9614

NEW MODIFIED LEFT-TO-RIGHT RADIX-R REPRESENTATION FOR INTEGERS

Arash Eghdamian[1*], Azman Samsudin[1]

[1]*School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia*

(Received: November 2016 / Revised: March 2017 / Accepted: March 2017)

ABSTRACT

This research addresses the problem of finding a minimum Hamming Weight by proposing a left-to-right recoding of integers (from the most significant bit to the less significant one). This representation is the enhanced and modified version of a well-known recoding method called ***G***eneralized ***N***on-***A***djacent ***F***orm (G-NAF). Scanning the digits from the left-to-right is called ***M***odified ***G***eneralized ***S***igned ***D***igit ***N***on-***A***djacent ***F***orm (MGSDNAF), which unlike the G-NAF, presents the 'nice property' to be obtained. A 'nice property' is one that is based on intuition and is particularly desirable to be obtained in a given context. This processing direction is of great importance because a table of pre-computed values may be used to speed up the scalar multiplication only for that direction. A subsequent advantage is that recoding the exponent in advance is not required. This results in better performances in both running time and memory space. This representation method can reduce the Hamming Weight of integers from about 21.6% for radix 3 to 15.1% for radix 9. These numbers for G-NAF recoding are 16.7% and 8.9% respectively. Comparing these numbers together shows that efficiency of the proposed method in reducing the Hamming Weight is more than the efficiency of G-NAF, which is from 30% (for radix 3) to more than 65% (for radix 9) more efficient in reducing the Hamming Weight. Finally, two radix 3 single scalar multiplication methods for Elliptic Curve Cryptography (ECC), which are based on G-NAF and Left-to-Right MGSDNAF, are compared in order to examine the application of the proposed method in cryptography. The results show that the proposed method can reduce the number of underlying arithmetic operations in single scalar multiplication by 14.1% while G-NAF can only reduce this number by 11.5%.

Keywords: Cryptography; Elliptic curve; Generalized NAF (G-NAF); Hamming weight; Radix-r representation

1. INTRODUCTION

Modular exponentiation is one of the most time-consuming operations in most of cryptosystems (Hankerson et al., 2000). Therefore, improving the efficiency of the algorithm which performs this operation is very important due to its direct impact on the performance of the resulting protocol of cryptography. To compute g^m (g raised to the power of m) basically two main types of exponentiation might be basically distinguished. In one type, the base (g) is fixed and the exponent (m) varies (e.g. ElGamal cryptosystems) (ElGamal, 1985; Menezes et al., 1997) while in the other type, there is a variable base (g) and a fixed exponent (m) (e.g. RSA cryptosystems) (Rivest et al., 1978; Koc, 1994). The current study is mainly concerned with the first type of exponentiation. Further use of this method is demonstrated when inverses can be computed

*Corresponding author's email: ae11_com109@student.usm.my, Tel: +604-653-3888, Fax: +604-657-3335
Permalink/DOI: https://doi.org/10.14716/ijtech.v8i3.6570

(virtually) for free (e.g. elliptic curves) (Morain & Olivos, 1990). The main idea is to enhance the effectiveness of multiplication by decreasing the Hamming Weight of the exponent. This study places an emphasis on elliptic curves defined over binary fields in order to show the application of the proposed method in Elliptic Curve Cryptography.

One of the most well-known representations of integers is ***G***eneralized ***N***on-***A***djacent ***F***orm (G-NAF) which is known as an efficient representation for radix-*r* integers (Clark & Liang, 1973; Muir, 2007). Clark and Liang's algorithm to build a G-NAF scans the number from the least significant digit to the most significant digit. The Hamming Weight (average density of non-zero digits) of this formation is asymptotically $\frac{(r-1)}{(r+1)}$ with (*r*-1) pre-computed points. For example, the Hamming Weight of the standard radix-*r* representation is $\frac{(r-1)}{r}$, so for $r = 3$ this number would be 0.67 with 2 pre-computed points. In the same radix 3, G-NAF has the average of 0.5 non-zero density with the same amount of pre-computed points. For that reason, the G-NAF can improve the efficiency of computing the paring based cryptosystem by making the scalar multiplication more efficient. Moreover, the non-zero density can be also reduced by adding new digits to the digit set and making the digit set larger.

Computing the scalar multiplication in Elliptic Curve Cryptography (ECC) is a good example. ECC requires computation of *mP* in the following form shown in Equation 1:

$$mP = \underbrace{P + P + \cdots + P}_{m\ times} \tag{1}$$

In Equation 1, *m* is an integer and *P* is a point on an elliptic curve. The two most important algorithms for computing *mP* are illustrated in Figures 1a and 1b, (Muir, 2007).

In both algorithms, a signed radix-*r* representation of the number *m* would be used instead of its normal presentation. The main difference of these two algorithms is the way that they process the digits. The algorithm shown in Figure 1a starts the process of the digits of *m*, from the less significant digit (right-to-left approach) and the one shown in Figure 1b performs the process from the most significant digit of *m* (left-to-right approach).

$$\begin{array}{l}
\text{for } i = 1 \cdots r-1 \\
\quad \text{do } P_i \leftarrow \infty \\
Q \leftarrow P \\
\text{Compute } (a_{l-1} \cdots a_1 a_0)_r = m \\
\text{for } i = 0 \cdots l-1 \\
\quad \text{do} \begin{cases} \text{if } a_i \neq 0 \\ \quad \text{then} \begin{cases} \text{if } a_i > 0 \\ \quad \text{then } P_{a_i} \leftarrow P_{a_i} + \mathrm{Q} \\ \quad \text{else } P_{-a_i} \leftarrow P_{-a_i} - \mathrm{Q} \end{cases} \\ Q \leftarrow rQ \end{cases} \\
Q \leftarrow \infty, R \leftarrow \infty \\
\text{for } i = 0 \cdots r-1 \\
\quad \text{do} \begin{cases} Q \leftarrow Q + P_i \\ R \leftarrow R + Q \end{cases} \\
\text{return } R
\end{array}$$

Figure 1a Right-to-Left radix-*r* method of computing *mP* (Muir, 2007)

$$\begin{array}{l}
Q \leftarrow \infty \\
\text{for } i = 1 \cdots r-1 \\
\quad \text{do} \begin{cases} Q \leftarrow Q + P \\ P_i \leftarrow Q \end{cases} \\
Q \leftarrow \infty \\
\text{Compute } (a_{l-1} \cdots a_1 a_0)_r = m \\
\text{for } i = l-1 \cdots 0 \\
\quad \text{do} \begin{cases} Q \leftarrow rQ \\ \text{if } a_i \neq 0 \\ \quad \text{then} \begin{cases} \text{if } a_i > 0 \\ \quad \text{then Q} \leftarrow \mathrm{Q} + P_{a_i} \\ \quad \text{else } \mathrm{Q} \leftarrow \mathrm{Q} - P_{a_i} \end{cases} \end{cases} \\
\text{return } Q
\end{array}$$

Figure 1b. Left-to-Right radix-*r* method of computing *mP* (Muir, 2007)

Since these two algorithms are not merely limited to some specific signed radix-*r* representations of *m*, there is no specific limitation in using other representation method. However, choosing a minimal weight representation can improve the algorithms' efficiency. The reason is that for each non-zero digit in *m*, an elliptic curve addition (or subtraction) should be performed, so less Hamming Weight means a fewer number of additions (or subtractions). The algorithm which performs from left-to-right (Figure 1b), is generally preferred because in this approach, if a pre-computed *mP* is required for different values of *m*, then the values of P_i, which will be set in the first loop, can be pre-computed and stored in advance.

Joye and Yen (2002) proposed a representation method which is carried out from left-to-right. Although their work has the same Hamming Weight result as G-NAF (Clark & Liang, 1973), the outputs are slightly different. In 2015, their method was used by Eghdamian and Samsudin to propose a modified left-to-right radix-*r* representation (Eghdamian & Samsudin, 2015a).

Later on, MGSDNAF (Eghdamian & Samsudin, 2015b) which was an improved version of G-NAF and MGNAF (Eghdamian & Samsudin, 2014) was presented. This method can represent radix-*r* integers with less Hamming Weight than a Generalized Non-Adjacent Form.

2. MGSDNAF

The proposed method by Eghdamian and Samsudin (2015b) is called ***M***odified ***G***eneralized ***S***igned ***D***igit ***N***on-***A***djacent ***F***orm or MGSDNAF. Since the main weakness of MGNAF is the size of its digit set, MGSDNAF was designed to reduce the size of MGNAF's digit set, while keeping it as efficient as the original MGNAF (Eghdamian & Samsudin, 2014). During MGNAF recoding process of an integer in radix r, a sequence of one non-zero digit (i.e. x) with length of *n* as shown in Equation 2:

$$(\cdots, y, \underbrace{x, x, \dots, x}_{n\ times}, \cdots) \tag{2}$$

will change as shown in Equation 3 to:

$$(y\frac{x}{r-1}, \underbrace{0,0,\dots,0}_{n-1\ times}\ \frac{x}{r-1}) \tag{3}$$

Based on MGNAF method the $(\frac{x}{r-1})$, would be added to the first digit, next to this sequence (left side) we consider as y. This addition results in the new digits in MGNAF's digit set. Therefore, the possibilities of $y\frac{x}{r-1}$ should be calculated in order to have the number of new extra digits created by MGNAF as shown in Equation 4:

For radix r:

$$x, y \in \{0,1,\dots,r-1\} \tag{4}$$

so $\frac{x}{r-1}$ can be r different numbers, but if $x = r-1$, then $\frac{x}{r-1}$ would become 1, and adding 1 to any digit of the G-NAF's digit set will not create any new digit which is not already in this set. In addition, if $x = 0$, $\frac{x}{r-1}$ would become 0, no new digit will be created by $x = 0$. For that reasons, $\frac{x}{r-1}$ will have $r-2$ new results and these results would be added to a digit which is y in Equation 2. The y ranges from 0 to $r-1$ and because y is the first non-x digit next to the

sequence of Xs, x should be excluded from this range, so y can obtain $r-1$ different values. In conclusion, the number of new digits in the digit set can be calculated based on the possible values of y multiplied by possible values of $\frac{x}{r-1}$. Therefore, $(r-1)(r-2)$ new digits would be created for the MGNAF algorithm for radix r in comparison with G-NAF.

To reduce the digit-set size, the third step of MGNAF recoding method is modified and improved. In the proposed algorithm (MGSDNAF), after recoding a sequence of non-zero digits and adding $(\frac{x}{r-1})$ to the next digit *y*, the newly created digit $(y\frac{x}{r-1})$ will be checked in order to see whether it is greater than $\frac{r}{2}$ or not. If so, the number will be deducted by r (i.e. radix) and the next digit will be increased by 1. With this new step, the number of extra digits in the digit set of MGSDNAF will be half of the numbers in MGNAF.

For example, the following sequence (5) can be a part of an integer number in radix *r* as shown in Equation 5:

$$(z, y\underbrace{x, x, \dots, x}_{n})_r \tag{5}$$

as it was mentioned before, the sequence (5) in MGNAF would be recoded to sequence (6) as shown in Equation 6:

$$(z, y\frac{x}{r-1}, \underbrace{0,0, \dots, 0}_{n-1\ times}, \frac{x}{r-1})_r \tag{6}$$

This is considered as the last step for MGNAF. After that, the next digit (z) would be checked for possible recoding while in MGDSNAF, the newly created digit $(y\frac{x}{r-1})$ would be checked and if it is greater than $\frac{r}{2}$, then the sequence (6) would be recoded to sequence (7) as shown in Equation 7:

$$(z+1, \left(y\frac{x}{r-1}\right)-r, \underbrace{0,0, \dots, 0}_{n-1\ times}, \frac{x}{r-1})_r \tag{7}$$

In terms of Hamming Weight, the experimental results show that the proposed method has similar Hamming Weight to MGNAF which is lower than G-NAF (see Table 1).

Table 1 Hamming Weight comparison of G-NAF and the proposed method

Radix	G-NAF	MGSDNAF
2	33.33	33.33
3	50	45.09
4	60	53.38
5	66.67	59.63
6	71.43	64.61
7	75	68.39
8	77.78	71.47
9	80	73.78

3. THE PROPOSED WORK

Joye and Yen (2000) proposed a new recoding algorithm for radix 2 and higher. In fact, it was an extension of their own left-to-right NAF recoding algorithm (Joye & Yen, 2000). In this paper, the idea of Joye and Yen's method (Joye & Yen, 2002) is adopted in order to propose a new left-to-right method for Modified Signed Digit Generalized Non-Adjacent Form (i.e. MGSDNAF (Eghdamian & Samsudin, 2015b).

In the proposed algorithm, variable *n* indicates the length of the integer *M*. Moreover, b_i shows the value of carry (i.e. borrow) that would be passed from the *i*-th digit *M* (which is shown as a_i) to the next digit (a_{i+1}). So the amount of the *i*-th digit of the new representation of *M*, is calculated based on b_i (the carry that it will pass to the next digit), b_{i-1} (the carry that it will receive), and its current value.

Algorithm: Given integer $M = \sum_{i=0}^{n} a_i r^i$, $|a_i| < r$, $i = 0,1,\cdots,n$, this algorithm computes an integer representation with the same Hamming Weight as MGSDNAF for *M* in three main steps.

***Step 1*:** Set $a_n = 0$; Set $b_n = 0$; Set $a_{-1} = 0$; Set $a_{-2} = 0$; Set $temp = 0$; Set $flag = 0$; Set $i = n$;

***Step 2*:** Do *step 3* while $i \geq 0$; then the algorithm terminates with $M' = \sum_{i=0}^{n} a'_i r^i$ as the new representation for *M*.

***Step 3*:** If $flag = 1$, then

If $a_{i-1} = temp$, then $b_{i-1} = \frac{temp}{r-1}$ and $i = i - 1$;

Else $b_{i-1} = 0$ and $i = i - 1$;

Else consider the following cases.

1) $a_{i-1} = a_{i-2}$**:** Set $temp = a_{i-1}$, Set $flag = 1$, Set $b_{i-1} = \frac{temp}{Radix-1}$ and Set $i = i - 1$
2) $a_{i-1} \geq \frac{r}{2}$ & $a_{i-2} = a_{i-3}$**:** Set $b_{i-1} = 1$ and Set $temp = a_{i-1}$
3) $a_i + a_{i+1} < (r-1)$**:** Set $b_{i-1} = 0$ and Set $temp = a_{i-1}$
4) $a_i + a_{i+1} = (r-1)$**:** If $b_i = 1$, $a_{i-1} = (r-1-temp)$ and $a_{i-1} + a_{i-2} < (r-1)$ then Set $b_{i-1} = 0$; Set $b_{i-1} = 1$ if $b_i = 0$, $a_{i-1} = (r-1-temp)$ and $a_{i-1} + a_{i-2} \geq r$; otherwise Set $b_{i-1} = b_i$
5) $a_i + a_{i+1} \geq r$**:** Set $b_{i-1} = 1$ and Set $temp = a_{i-1}$
6) End case and Set $a'_i = -rb_i + a_i + b_{i-1}$

4. RESULTS AND DISCUSSION

Similar to the Left-to-Right MGNAF, this study also uses the main idea of Joye and Yen's work in (Joye & Yen, 2002) to propose a new left-to-right method for MGSDNAF recoding. This left-to-right algorithm also finds the same non-zero digit sequences and recodes them with new method which is the modified version of Joye and Yen's (2002) work. The result would be similar to the MGSDNAF recoding. The rest digits would be transformed with the same method as proposed in Joy and Yen's method.

For example:

$$M = (\dots, a_q, a_{q-1}, a_{q-2}, x, x, x, x, a_l, a_{l-1}, a_{l-2}, \dots) \quad (8)$$

In this example (8): all digits are in set of $\{0,1,\cdots,r-1\}$ and $a_{q-2} \neq x$ and $a_l \neq x$. In addition, four Xs made the same non-zero digit sequence with a length of four.

The MGSDNAF algorithm starts from right (the least significant digit) and proceeds in the same way that G-NAF and MGNAF do until reaching the sequence of Xs. In that position, MGSDNAF recodes this sequence with a method similar to MGNAF. In the next step, after recoding the sequence of Xs, the MGSDNAF method checks whether the next digit is greater than half of the radix or not. If it is so, then the value of the radix would be deducted from that digit and the next digit would be increased by 1. Then, the MGSDNAF continues with other digits with the same method used in G-NAF and MGNAF until reaching the next same non-zero digit sequence. So the number can be divided into three parts as follows in Equations 9 and 10:

$$M = \cdots, a_q, a_{q-1}, a_{q-2}, \underset{x,x,x,x}{} \underset{a_l, a_{l-1}, a_{l-2}, \cdots}{} \tag{9}$$

and the middle part would be changed to:

$$\frac{x}{r-1}, 0,0,0, \frac{-x}{r-1} \tag{10}$$

in which $\frac{x}{r-1}$ would be added to a_{q-2}. Now, if $(a_{q-2} + \frac{x}{r-1})$ is greater than $\frac{r}{2}$, then a_{q-2} would be reduced by r and a_{q-1} would be increased by 1. The Hamming Weight of the first and the third part would be the same as G-NAF and MGNAF, but transforming the same non-zero digit sequence with MGSDNAF method is the key to better Hamming Weight of this representation algorithm.

In the same example, the left-to-right proposed method starts from left-to-right and proceeds as the method of Joye and Yen does until reaching the digit exactly before the same non-zero digit sequence. So this method checks two digits ahead. In this case, if they are equal, it means the current digit is the digit before the sequence and if this digit is greater than $\frac{r}{2}$, b_i would be set as 1 (it means that this digit would be deducted by *r*). Then, it stores the value of the first digit of the sequence (i.e. x) in $temp$. From that point, the proposed method would set the carry for the next digit as $\frac{temp}{r-1}$ (i.e. $b_{i-1} = \frac{x}{r-1}$), as long as the next digit has the same value as $temp$. For example, the second *x* in the sequence would be recoded to:

$$a'_i = -rb_i + a_i + b_{i-1} \tag{11}$$

Since it is the second x, both b_i and b_{i-1} are $\frac{x}{r-1}$. So the result would be:

$$\begin{aligned} a'_i &= -rb_i + a_i + b_{i-1} \\ &= -r\frac{x}{r-1} + x + \frac{x}{r-1} \\ &= (1-r)\frac{x}{r-1} + x \\ &= -x + x \\ a'_i &= 0 \end{aligned} \tag{12}$$

In conclusion, since the Hamming Weight of the same non-zero digit sequences would be similar to MGSDNAF and the rest would be similar to Joye and Yen's method (Joye & Yen 2002), the total Hamming Weight of the proposed algorithm would be similar to MGSDNAF.

5. APPLICATION OF THE PROPOSED METHOD

To check the application of the proposed method in cryptography, a single scalar multiplication based on left-to-right MGSDNAF has been studied. The first step of this study was to calculate the frequency (percentage of appearance) of each digit related to the total number of digits in a number. This study has been done for normal presentation, G-NAF and the proposed method.

The second step was calculating the number of operations in a simple single scalar multiplication process for each representation followed by comparison of these representations. This study only focuses on analysing the impact of Hamming Weight reduction on the number of total simple operations in a single scalar multiplication process in these methods. Therefore, no customization has been used to improve the operations. For sure, improving the operations or using some additional methods like windowing, using lookup tables or even using double base or multi base numbering systems can be helpful to increase the efficiency of the multiplication, but they might affect the results of the Hamming Weight reduction analysis.

To apply the proposed method in ECC, radix 3 was chosen for the key size of 256 bits. Applying new methods for radixes higher than 3 in ECC might be more effective, however it needs more operations to be defined, while all needed operations for radix 3 are already available. About the key size, 256 bits is the current standard key size for Elliptic Curve Cryptography. Therefore, 1000 different random numbers with length of 256 bits were generated and all the results are the average results for these 1000 numbers.

Then, the number of each digit of the digit set, ignoring their sign (for this example $\left\{0, \frac{1}{2}, 1, 2\right\}$) for normal representation were counted and the frequency of each digit related to the total number of digits were calculated. After that, each of these 1000 random numbers were recoded with both methods, namely G-NAF and left-to-right MGSDNAF and same studies were applied on them again. The results are shown in Table 2.

Table 2 The frequency (percent of appearance) of each digit related to the total number of digits

Digit	Normal (%)	G-NAF (%)	Left to Right MGSDNAF (%)
0	33.3	50	55
1/2	0	0	13.5
1	33.3	33.5	20.5
2	33.3	16.5	11

In all implementation of ECC primitives, scalar multiplication is the computationally dominant operation. Several methods have been proposed to speed-up point multiplication. These methods use various representations of the base point (e.g. affine coordinates, projective coordinates), various representations of the scalar (e.g. binary, ternary, NAF, w-NAF), and various curve operations (e.g. additions, doublings, halvings, triplings) (Ciet et al., 2006). The computational cost (timing) of these curve operations depends on the cost of the arithmetic operations that have to be performed in the underlying field. In general, addition and subtraction in the underlying field are operations of negligible cost.

First, randomly generated integer numbers were recoded with each method and the number of each digit of these numbers were counted. Then, the number of arithmetic operations used in a single scalar multiplication of an Elliptic Curve Cryptography process was calculated. To be more precise, the number of inversions, squarings and multiplications (that denote by [i], [s]

and [m], respectively) was taken into account for scalar multiplication based on normal presentation, G-NAF and the proposed left-to-right MGSDNAF.

Table 3 Number of inversions [i], squarings [s] and multiplications [m], for different curve operations over F_{2^m} using affine coordinates

Curve operation	Binary field
$\frac{1}{2}P$	$2[m]$
$\frac{1}{2}P + Q$	$1[i] + 1[s] + 4[m]$
$P + Q$	$1[i] + 1[s] + 2[m]$
$2P$	$1[i] + 1[s] + 2[m]$
$2P + Q$	$1[i] + 2[s] + 9[m]$
$3P$	$1[i] + 4[s] + 7[m]$
$3P + Q$	$2[i] + 3[s] + 9[m]$

The choice of parameters used for a specific implementation mostly depends upon the ratio, [i]/[m], between one inversion and one multiplication. In binary fields, it is assumed to be between 3 and 8 (Hankerson et al., 2000), whereas in prime fields it is between 30 and 50 (Fong et al., 2004). Furthermore, a squaring [s] is generally assumed to be roughly equal to 0.8[m] in prime fields and almost free in binary fields (see (Hankerson et al., 2000) for more details). Table 3 demonstrates the costs of the curve operations which are used in analysis. More details can be found in (Eisentrager et al., 2003; Ciet et al., 2006).

Therefore, based on the number of operations (Table 2) and their costs (Table 3), the cost of doublings and additions in ECC process for each method is calculated under the assumptions that a field division costs roughly the same as inversion and $i = 8m$, s = free and Halving = $2m$. As it is shown in Table 4, left-to-right MGSDNAF can increase the number of "deducted underlying arithmetic operations" from 276 for G-NAF to 337. This means that left-to-right MGSDNAF recoding can decrease the number of operations needed for a single scalar multiplication at about 14.1% in comparison to normal presentation.

Table 4 Number of curve and field operations for 100-bit example in binary field

	Normal				G-NAF				Left-to-Right MGSDNAF			
Operations	[*i*]	[*s*]	[*m*]	≈Total*	[*i*]	[*s*]	[*m*]	≈Total	[*i*]	[*s*]	[*m*]	≈Total
Additions	66	99	363	891	50	66.5	215.5	615	45	56	194	554
Triples	100	400	700	1500	100	400	700	1500	100	400	700	1500
Total	166	499	1063	2391	150	466	915	2115	131.5	442	1002	2054

*All operations converted to [*m*]

6. CONCLUSION

This paper introduced a new algorithm for representing the radix-r integers with the intention of reducing the hamming weight and improving the efficiency of point multiplication in pairing-base cryptosystems. In the proposed algorithm, the integer numbers will be recoded from left-to-right (from the most significant digit to the list significant one) to a new form which has the same Hamming Weight as MGSDNAF. This approach is beneficial in terms of decreasing time and space complexities of some public-key cryptosystems such as ECC. Finally, the application

of the proposed algorithm on single scalar multiplication in Elliptic Curve Cryptography over binary fields was examined.

7. REFERENCES

Ciet, M., Joye, M., Lauter, K., Montgomery, P.L., 2006. Trading Inversions for Multiplications in Elliptic Curve Cryptography. *Designs, Codes, and Cryptography*, Volume 39(2), pp. 189–206

Clark, W., Liang, J., 1973. On Arithmetic Weight for a General Radix Representation of Integers. *IEEE Transactions on Information Theory*, Volume 19, pp. 823–826

Eghdamian, A., Samsudin, A., 2015a. A Modified Left-to-right Radix-r Representation. *In*: 2015 International Symposium on Technology Management and Emerging Technologies (ISTMET), pp. 254–257

Eghdamian, A., Samsudin, A., 2014. An Improved Signed Digit Representation of Integers. *In*: 3rd International Conference on Computer Engineering & Mathematical Sciences, pp. 287–290

Eghdamian, A., Samsudin, A., 2015b. MGSDNAF - A Modified Signed Digit Generalized Non- Adjacent Form for Integers Representation. *In*: 2nd Advancement on Information Technology International Conference (ADVCIT)

Eisentrager, K., Lauter, K., Montgomery, P., 2003. Fast Elliptic Curve Arithmetic and Improved Weil Pairing Evaluation. *Topics in Cryptology—CT-RSA 2003*, pp. 343–354

ElGamal, T., 1985. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory*, Volume 31(4), pp. 469–472

Fong, K., Hankerson, D., Lopez, J., Menezes, A., 2004. Field Inversion and Point Halving Revisited. *IEEE Transactions on Computers*, Volume 53(8), pp. 1047–1059

Hankerson, D., Hernandez, J.L., Menezes, A., 2000. Software Implementation of Elliptic Curve Cryptography over Binary Fields (Invited Talk). *In*: Cryptographic Hardware and Embedded Systems, Springer, pp. 1–24

Joye, M., Yen, S.M., 2002. New Minimal Modified Radix-r Representation with Applications to Smart Cards. *In*: Naccache D., Paillier P. (eds) Public Key Cryptography. Lecture Notes in Computer Science, Volume 2274, pp. 375–383. Springer, Berlin, Heidelberg

Joye, M., Yen, S.M., 2000. Optimal Left-to-right Binary Signed-digit Recoding. *IEEE Transactions on Computers*, Volume 49(7), pp. 740–748

Koc, C.K., 1994. *High-speed RSA Implementation*. Technical Report, RSA Labratories, November

Menezes, A., Oorschot, P., Van & Vanstone, S., 1997. *Handbook of Applied Cryptography*, CRC Press

Morain, F., Olivos, J., 1990. Speeding Up the Computations on an Elliptic Curve using Addition-subtraction Chains. *Informatique théorique et Applications*, Volume 24(6), pp. 531–543

Muir, J.A., 2007. A Simple Left-to-right Algorithm for Minimal Weight Signed Radix-r Representations. *IEEE Transactions on Information Theory*, Volume 53(3), pp. 1234–1241

Rivest, R., Shamir, A., Adleman, L., 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, Volume 21(2), pp. 120–126