N.M.Ghanim
Branch of Applied Mathematics-Applied Sciences Department
University of Technology-Baghdad-Iraq
nadiamg08@gmail.com

M. A. Magamiss
Branch of Applied Mathematics-Applied Sciences Department
University of Technology-Baghdad-Iraq
m.altemimey@gmail.com

Received on: 19/05/2016
Accepted on: 19/01/2017

Nested Method for Optimizing Elliptic Curve Scalar Multiplication

Abstract-*The algebraic curve that attracted considerable interest in recent years is called Elliptic Curve (EC). This is due to the computational complexity of its arithmetic over a finite field. The complexity of its arithmetic operations granted EC highly interest for many applications, especially in Cryptography. The scalar multiplication plays an important role in the performance of the elliptic curve cryptosystem (ECC). This paper focused on optimizing the performance of this important operation, which is called elliptic curve scalar multiplication (ECSM). As known from previous works, this operation can be sped up using one of the most important representations called Mutual Opposite Form (MOF). Based on this representation, we proposed an algorithm to improve the performance of ECSM. The efficiency of the proposed algorithm is enhanced in terms of computation time compared to the existing standard ECSM methods.*

Keywords- *Elliptic curve scalar multiplication (ECSM), Binary method, Mutual opposite form (MOF).*

How to cite this article: N.M.G,M. A. Magamiss, “Nested Method for Optimizing Elliptic Curve Scalar Multiplication,” *Engineering and Technology Journal*, Vol. 35, Part B, No. 2, pp. 105-111, 2017.

1. Introduction

The concept of the elliptic curve cryptography *ECC* was introduced by Victor Millar [1] and Neal Koblitz [2] in 1986 and 1987 independently. *ECC* appears strongly compared with other existing public key cryptosystems like RSA [3], DSA [4] and DH [5], due to its shorter key size and high performance. the security level for key of length 160 bits in ECC is equivalent to RSA security with key of length 1024 bits, DSA and DH [6]. These advantages make *ECC* more important to be utilized in the limited environments such as; smart carts, wireless sensor network, and PDA [7-9]. Since introducing *ECC* by Miller and Koblitz, many researchers have devoted their efforts to improve its efficiency [10-12].

The responsible of the security in *ECC* is the discrete logarithms problem (*DLP*) which is concerned on how to find the integer k for two given points P_2 and P_1 on *EC*, such that $P_2 = kP_1$, k is known as a scalar in elliptic curve cryptography and represents the secret key. Elliptic curve scalar multiplication can be expressed as,

$$kP_1 = \underbrace{(P_1 + P_1 + \cdots + P_1)}_{k-times}$$

which is the dominant operation in *EC*. It is controlled the cost and the time, which motivated many researchers to study this problem and proposed their methods to speed-up its computation. Some known and mostly used methods are the binary and the signed binary representation as complementary recoding [13-15], *NAF* [16-19], *MOF* [20], and *DRM* [21]. Two operations determine the speed of *ECSM;* they are the elliptic curve adding (*ECADD*) and doubling (*ECDBL*) operations. The number of bits and the number of hamming weight for the scalar k in the signed binary representation determines the number of (*ECADD*), and (*ECDBL*). The *ECC* defined over two fields. The first is a prime field and the other is a finite field. In this work, the proposed method is defined over the prime field. The main contribution in this paper focused on optimizing the computation time for *ECSM* based on the most using signed binary representation, the mutual opposite form (*MOF*) [20].

This paper is organized in five more Sections: Elementary properties of the elliptic curve with its basic operations are discussed in Section 2. Section 3 presents some well-known proposed methods for improving of *ECSM* algorithm. Section 4 introduced the proposed algorithm with implementation. Some results and complexity analysis are explained in section 5, and finally, the paper is concluded in section 6.

2. Literature Review

In 2015, H. Almimi et al. [22] proposed *ZOTEC* method by a method known as *ZOTEC* based on *ZOT* recording method for speeding of the EC

https://doi.org/10.30684/etj.2017.138621
2412-0758/University of Technology-Iraq, Baghdad, Iraq

process. This method can be applied left to right or right to left. They proved that their method is more efficient that *NAF*, *MOF* and *CR* when the binary operation is included or excluded .They also proved that ,the *ZOTEC* method is efficient than a for mentioned method sin terms of time and space.

In 2015, N. AL-SAFFAR [23] accelerated The *ECSM* based on *w*-nonadjacent form method. She reduced the number of operation in *ECSM*. The improvement percentage is about 20% over the binary method and 14% over *NAF* and 7.6% over *w*-NAF.In 2014, A Mohammad et al. [6] optimized the *ECSM* based on *MOF* method. Their method combines the add-subtract al algorithm of the scalar multiplication with the *MOF* method. They achieve a 90%speed up in comparing to the existing methods. They proved by implementation that their method is efficient and produced good reduction in the computational time.In 2012, A, Rezai and P. Keshavarzi [24] proposed an algorithm to convert the integer from the binary representation to the complementary canonical sliding window (*CCS*) representation by using the complementary method. The canonical recoding method and the sliding window method consecutively. He was proved that the average Hamming weight of the *CCS* representation by using Markov chain is $\frac{39n}{39w+80}$ for n-bit integer with window width w. His analysis explained that the average Hamming weight of the *CCS* representation is reduced compared to other representations. Thus, utilizing the *CCS* representation in the scalar multiplication, the average number of the point addition/subtraction operation is reduced compared to other scalar multiplication algorithms considerably.In 2012, S. Vorapong and H. Imai [25] proposed an algorithm for producing the double-base chain to improve the time used for evaluating an elliptic curve scalar multiplication. His algorithm is the first to achieve the minimum time by using dynamic programming. Compared with greedy-type algorithm, the experiments show that his algorithm reduced the time for evaluation the scalar multiplication by 3.88-3.95% with almost the same average running time for the method itself. Also, they extend their idea, and proposed an algorithm to optimize multi-scalar multiplication. By that extension, they achieved an improvement for computation time of the operation by 3.2-11.3%.In 2012, N Shylashree and V. Sridhar [26] introduced a new method by using Ancient Indian Vedic for hastening scalar multiplication in elliptic curve cryptography. Their proposed work is six times faster than the previous work when applied in point doubling using Spartan3 as target device.

3. Preliminaries

This section presents an overview of the material used in this work, for more details and some background in cryptography; we refer the reader to see [27-29].

Definition 3.1: Let *F* be any field. Based on Weierstras's equation, the elliptic curve *E* over *F* is defined by:

$$y^2+a_1xy+a_3\,y=x^3+a_2x^2+a_4x+a_6 \qquad (1)$$

where, the coefficients $a_i \in F$ and the discriminant of *E is* $\Delta \neq 0$.

The discriminant Δ of E given in (1) is defined by;

$$\Delta=-\,d_2^{\,2}\,d_8-8d_2^{\,3}-27\,d_2^{\,2}+9d_2d_4d_6 \qquad (2)$$

where

$d_2=a_1^{\,2}+4\,a_2$

$d_4=2a_4+\,a_1\,a_3$

$d_6=a_3^{\,2}+4\,a_6$

$d_8=a_1^{\,2}\,a_8+4\,a_2\,a_6-\,a_1a_3a_4+a_2\,a_3^{\,2}-a_4^{\,2}$

Let *L* be an arbitrary extension of *F*, the set of all points on *E* is given by:

$E(L)=\{(x,y)\in L\times L{:}\ y^2+a_1\,xy+a_3\,y-x^3-a_2x^2-a_4\,x-a_6=0\}\cup\{0_\infty\}$.

where 0_∞ is called the point at infinity.

The expression of the Weierstrass equation given in (1) for elliptic curve over the prime field F_p can be simplified as follows:

$d_8=a_1^{\,2}\,a_8+4\,a_2\,a_6-\,a_1a_3a_4+a_2\,a_3^{\,2}-a_4^{\,2}$

Let *L* be an arbitrary extension of *F*, the set of all points on *E* is given by:

$E(L)=\{(x,y)\in L\times L{:}\ y^2+a_1\,xy+a_3\,y-x^3-a_2x^2-a_4\,x-a_6=0\}\cup\{0_\infty\}$.

where 0_∞ is called the point at infinity. The expression of the Weierstrass equation given in (1) for elliptic curve over the prime field F_p can be simplified as follows:

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

where $a,\ b\ \in R$ and $\Delta=4a^3+27b^2\neq 0$ [19].

I. Addition of two points over *EC* algebraically

If we have two different points $P_1=(x_1,\ y_1)$, and $P_2=(x_2,y_2)\in E(F_p)$, then its addition operation defined as $P_1+P_2=(x_1,\ y_1)+(x_2,y_2)=(x_3\,,y_3)=P_3\in E(F_p)$ is computed as follows:

$$P_1+P_2=\begin{cases} O\infty\ \ if\ x_1=\ x_2\,y_1=-y_2 \\ \ (x_3,y_3)\ \ \ if\ otherwise\end{cases} \qquad (4)$$

where

$$x_3=\lambda^2-x_1-x_2 \qquad (5)$$

$$y_3=\lambda\,(x_1-x_3)-y_1 \qquad (6)$$

and λ is the slope of the line *L* that connecting the points $P_1=(x_1,\ y_1)$ and $P_2=(x_2,y_2)$ such that:

$\lambda = [(y_2 - y_1)/(x_2 - x_1)] \; mod \; p$ (7)

If we have two similar points $P_1 = (x_1, y_1) \in E(F_p)$ and itself, then its addition operation with itself is defined as $P_1 + P_1 = (x_1, y_1) + (x_1, y_1) = (x_3, y_3) = 2P_1 \in E(F_p)$. It is computed as follows:

$x_3 = \lambda^2 - x_1 - x_1$ (8)

$y_3 = \lambda (x_1 - x_3) - y_1$ (9)

where

$$\lambda = \frac{3{x_1}^2 + a}{2y_1} \quad (10)$$

The inverse of the point $P_1 = (x_1, y_1) \in E(F_p)$ is defined as $-P_1 = (x_1, -y_1) \in E(F_p)$.

Definition 3.3: The number of non-zero bits in the signed binary representation is known as the hamming weight of the scalar *k* denoted by *h* (*k*).

Definition 3.2: A signed binary representation of a scalar k with the base b is $(k)_b$ such that $k = k_{l-1}k_{l-2} \dots k_0$ with $|k_i| < b$ for $i = 0, 1, 2, \dots, l-1$ and $k = \sum_0^{l-1} k_i \, b^i$.

Definition 3.3: the length of a singed binary representation denoted by $l(k)$ is the number of bits in a signed binary representation.

4. Previous work

The binary representation length of the scalar *k* and its number of 1's are controlling the efficiency and the cost of the *ECSM*. There has been extensive research toward achieving of an efficient representation for *k*. From literature, some of the existing well-known methods are presenting in this section.

I.Binary Method:

The simplest method to represent an integer in (0, 1) bits is called the binary representation. It is defined as $(k_{l-1} \; k_{l-2} \dots k_0)_2$ where $k_i \in (0, 1)$, $i = 0, 1, 2 \dots, l-1$. Thus any integer can be expressed as $k = \sum_0^{l-1} k_i \, 2^i$. Let P_1 be a point on an elliptic curve *E*, then the scalar multiplication defined as $kP_1 = \sum_0^{l-1} k_i \, 2^i \, P_1 = P_2$, where P_2 is another point on *EC*. The computation of the scalar multiplication for binary method is showing in Algorithm 1:

Algorithm 1: *ECSM* based on Binary Method

```
Input: $k = (k_0, \; k_1, \dots, k_{l-1})_2$, $P_1 \in E(F_p)$
$Output$: $P_2 = kP_1$
1.$P_2 = P_1$
2. for $i = l - 1$ down to 0 do
    2.1. $P_2 = 2P_1$
    2.2. $If \; k_i = 1 \; then \; P_2 = P_2 + P_1$
3. Return $P_2$.
```

The bits of *k* can be scans either from left to right or right to left in this method. The number of operations for execution this Algorithm is determined the running time. The elliptic curve adding (*ECADD*) and doubling (*ECDBL*) operations are performed if $k_i = 1$, otherwise, only elliptic curve doubling (*ECDBL*) operation is performed. The average of getting the number of ones in the binary expansion of the scalar *k* is $l - 1$.

II.Mutual Opposite Form (MOF)

In 2004, Mutual opposite form (*MOF*) proposed by Okeya et al. [20], is introduced to reduce the hamming weight by converting the scalar to signed binary representation. The integer with length l of the binary representation by *MOF* is at most of the length $(l + 1)$ and has a unique representation. *MOF* satisfies the following properties:

a- 1.The signs are different for every two adjacent non zero bits

b- 2.The values 1and -1 are the first and the last non-zero bit in *MOF* representation, respectively.

c- 3.Every positive integer can be expressed by a unique *MOF*.

d- 4.Sliding window method with width *w* can be executed in *MOF*, but in other methods such as binary complementary, it cannot be executed with the sliding.

To represent the integer *k* in *MOF* we follow

2k =	k_{l-1}	k_{l-2}	…	k_{r-1}	…	k_1	k_0	
Θk =		k_{l-1}	…	k_r	…	k_2	k_1	k_0
MOF(*k*)	k_{l-1}	k_{l-2} -k_{l-1}	…	k_{r-1} - k_r	…	k_1 -k_2	k_0 -k_1	-k_0

where Θ refers to bitwise subtraction.This calculation can be computed in two directions; left to right (*L2R*) or right to left (*R2L*), as presented in Algorithm 2 and 3 respectively.

The *L2R MOF* representation is presented in Algorithm 2:

```
Algorithm2: (L2R) MOF representation of a scalar k
Input: $k = (k_{l-1} \dots k_1 k_0)_2$
Output: $k = (d_l\ d_{l-1} \dots d_1 d_0)_{MOF(k)}$
    Set $d_l$=$k_{l-1}$
    For $i = l-1$ down to 1 do
        $d_i$=$k_{i-1}$ - $k_i$
    $d_0$= -$k_0$
    Return$(d_l\ d_{l-1} \dots d_1 d_0)_{MOF(k)}$
```

Example1: Let *k*=7, Table 1, present the steps for converting the scalar *k* into *L2R MOF* representation.

Table 1: Binary Representation for *k*=7

Iteration value (*i*)	**Binary representation**	***MOF*(7)**
3		1
2	1	10
1	11	100
0	111	100$\bar{1}$

The second *MOF* representation (*R2L*) is presented in Algorithm 3.

```
Algorithm3: (R2L) MOF representation of k
Input: $k = (k_{l-1} \dots k_1 k_0)_2$
Output: $k = (d_l\ d_{l-1} \dots d_1 d_0)_{MOF(k)}$
    Set $d_0$= -$k_0$
    For $i = 1$ down to $l-1$ do
        $d_i$=$k_{i-1}$ - $k_i$
    $d_l$=$k_{l-1}$
    Return$(d_l\ d_{l-1} \dots d_1 d_0)_{MOF(k)}$.
```

III.Elliptic Curve Scalar Multiplication (ECSM)

Let *P* be a point on the elliptic curve *E*, the number of adding of a point *P* with itself *k* time is known as elliptic curve scalar multiplication. If we have *MOF* representation, then ECSM based on *MOF* can be calculated in Algorithm 4.

```
Algorithm 4: L2R MOF to find ECSM
Input: $k = (d_l\ d_{l-1} \dots d_1 d_0)_{MOF(k)}, P_1 \in E(F_p)$
$Output: P_2 = kP_1$
$P_2$=0
for $i = l-1$ down to 0 do
    $P_2 = 2P_1$
    if $d_i = 1$ then $P_2 = P_2 + P_1$
    if $d_i = -1$ then $P_2 = P_2 - P_1$
Return $P_2$.
```

Table 2: ECSM by Aalgorithm 4

i	*MOF*(7)	*ECSM*	Operations
3	1	2*P*	Initialization
2	0	2(2*P*)	*ECADD*
1	0	2(2(2*P*))	*ECADD*
0	-1	2(2(2*P*))-*P*	*ECSUB*

5.

6.

5.The Proposed Algorithm

The proposed method combines Algorithm 2 or 3, and elliptic curve scalar multiplication, Algorithm 4 to produce a new elliptic curve scalar multiplication faster than those produced by some existing algorithms. The proposed algorithm and some of the well-known algorithms are carried out using MATLAB Ver. 7. The results are obtained using hardware with the specifications, CPU Intel 2.2GHz, Cor. i3 and 2 GB RAM.

I

I.Hybridization MOF with ECSM

First, we propose a new serial multiplication algorithm. It used for calculating the elliptic curve scalar multiplication based on the binary representation. To generate a signed binary representation *MOF*, we use Algorithm 2 or 3, then Algorithm 4 is applied to generate the elliptic curve scalar multiplication *kP*, where *P* is a point on *EC* and *k* is represented in mutual opposite form. To perform this task by the proposed algorithm, mutual opposite form is embedded implicitly in the elliptic curve scalar multiplication algorithm to produce new representation by single loop. We calculate the elliptic curve scalar multiplication directly when the scalar *k* is converted to the binary representation. Two possibilities for the binary scalar, that is either bit `0' or bit `1'. Already, we defined that ϴ is a bitwise subtraction. Now, the input for this algorithm is *l* the bit integer *k* =$(k_{l-1} \dots k_1 k_0)_2$. This representation is extended by adding two zeros, one to left k_l=0 and the other to right k_{-1}=0. After that a bitwise-subtraction ϴ between any two bits k_i - k_{i-1} is performed, for *i* =0, 1,…, *l*. Elliptic curve adding (*ECADD*) and elliptic curve doubling (*ECDBL*) are performed if the bitwise-subtraction k_i - k_{i-1} =1, for *i =0, 1,…, l*. Furthermore, Elliptic curve subtraction (*ECSUB*) and elliptic curve doubling (*ECDBL*) are performed if the bitwise-subtraction k_i - k_{i-1} = -1, for *i =0, 1,..., l*. This Algorithm 5 helps to reduce the processing time for this operation.

Algorithm 5: Nested *ECSM* and *MOF* algorithms

```
Input: $k = (k_{l-1} \dots k_1 k_0)_2$ , $P_1(x, y) \in E(F_p)$
Output: $P_2 = k P_1$
$P_2$=0;
$k_{-1}$ =0, $k_l$ =0;
For $i = l - 1$ to -1
If $k_i$ - $k_{i+1}$ = 1 then $P_2 = P_2 + P_1$
If $k_i$ - $k_{i+1}$ = -1 then $P_2 = P_2 - P_1$
$P_2 = 2P_2$
Return $P_2$.
```

Example 3: Let k = 23= $(10111)_2$ and $P \in E(F_p)$ be a point on *EC*, Algorithm 5 is implemented for this number as presented in Table 3.

6.Algorithm Analysis

Two processors are used for the proposed method one for adding and other for doubling. They are used for calculating of *ECSM* based on the binary representation. Based on the decomposition principle, the proposed algorithm is work, which means, the *ECSM* is calculated by employing of the two processors together. There for it is easy to conclude that, there is dependency between the adding and the doubling operations, i.e, the date comes from the doubling operations is used to execute the adding operation .Hence, to achieve a data independency, the task should be decomposed into two independent subtasks. This can be achieved by using of circular buffer for transmitting data between processors.The main two sections of the proposed algorithm, which are the *ECDBL* and the *ECADD* should be executed at the same time. It is so important to make sure that the *MOF* scalar is not zero. Before executing of the *ECADD* operation, the given point *P* is read by *ECDBL* processor before performing of the doubling operations. The proposed algorithm performance is measured. The execution time result of implementing *ECADD* and *ECDBL* operations is given in the second unit, as listed in Table 4 and Figure 1.

II I.Computational Complexity of Algorithm 5

Using hardware is a costly issue, efficient algorithms lead to an efficient use of the hardware. Studying the cost of solving the interesting problems is called computational complexity. It has two rescores; the running time, and the occupied space. For the proposed algorithm, the analysis shows that, by using the nested *ECSM* as a parallel mechanism, the computational complexity is reduced compared to some serial traditional. The required time for performing the *ECSM* is $O(n)$, whereas, using the traditional algorithm it was $O(n) * O(n)$ running amount of time. Therefore, the proposed algorithm is implemented faster; this can be obviously shown in Figure 1.

Table 3: *ECSM* by algorithm 5 for *k*=23

i	Binary representation	Nested *ECSM* with *MOF*	Operations
4	1	2*P*	*ECDBL*
3	0	2(2*P*-*P*)	*ECSUB and ECDBL*
2	1	2(2(2*P*-*P*)+*P*)	*ECADD and ECDBL*
1	1	2(2(2(2*P*-*P*)+*P*))	*ECDBL*
0	1	2(2(2(2(2*P*-*P*)+*P*)))	*ECDBL*
		2(2(2(2(2*P*-*P*)+*P*)))-*P*	*ECSUB*

Table 4: The Computational Time of Scalar Multiplication

Length of bits	Standard *ECSM*	Nested Algorithm	Speed up
52	0.0258	0.0204	1.26
128	0.0448	0.0344	1.30
256	0.0831	0.0338	2.46
512	0.4132	0.0557	7.42
630	0.5787	0.0568	10.19
886	1.2606	0.0717	17.58
1024	1.6960	0.0775	21.884

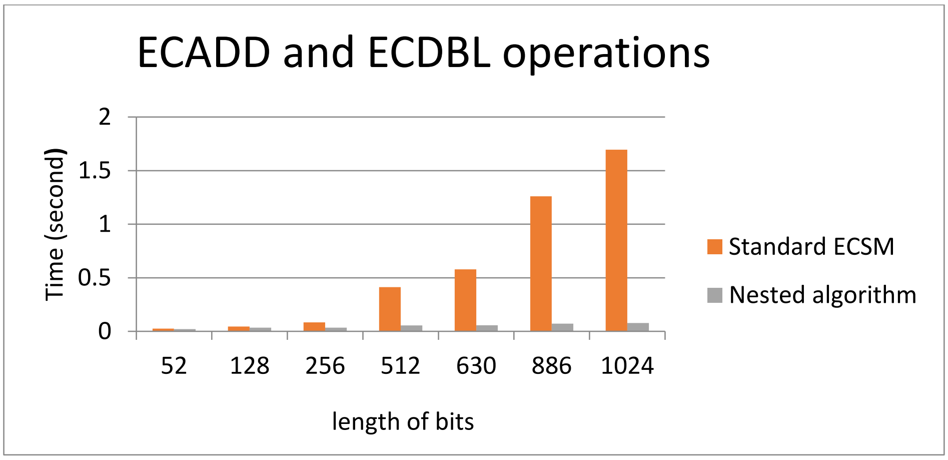


Figure1. The Computational Time of Scalar Multiplication

7.Conclusion

In *ECC*, the scalar multiplication is the most important operation, while, it is the most expansive operation due to the consuming of the time implementation. The speed improvement is a challenge that many researchers try to reach it. The performance of the scaler multiplication is mainly based on the representation of the scalar. In our work, we design and implement of a new efficient algorithm called the nested algorithm, which is combined the elliptic curve scalar multiplication and a signed binary representation *MOF* algorithm. The running time of the nested algorithm has been reduced compared with *ECSM* based on *MOF* representation according to the results.

References

[1].V.S. Miller, "Use of Elliptic Curves in Cryptography, Advances in Cryptology," Proceedings of CRYPTO.85, pp. 417-426, 1986.

[2].N. Koblitz, "Elliptic Curve Cryptosystem," Mathematics of Computation, Vol.48, pp.203- 209, 1987.

[3].L. Ronald Rivest, Adi Shamir, and Len Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, 21(2), pp.120–126, 1978.

[4].G. Locke and P. Gallagher, "Digital Signature Standard (DSS)," Federal information processing standards publication. National Institute of Standards and Technology, 2009.

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

[6].M. Anagreh, A. Samsudin and M .A Omar, "Parallel Method for Computing Elliptic Curve Scalar Multiplication Based on MOF," The International Arab Journal of Information Technology, 11(6), 2014.

[7].Y. Dan, X. Zou, Z. Liu, Y. Han, and L. Yi, "High-Performance Hardware Architecture of Elliptic Curve Cryptography Processor over GF (2^{163})," J. Zhejiang Univ. Sci. A, 10(2), pp.301-310, 2009.

[8].A. Osmani, "Design and Evaluation of New Intelligent Sensor Placement Algorithm to Improve Converge Problem in Wireless Sensor Networks," J. Basic. Appl. Sci. Res., 2(2), pp.1431-1440, 2012.

[9].I. Blake, G. Seroussi, and N. Smart, "Elliptic Curves in Cryptography," Cambridge University Press, UK, 1999.

[10].E. Al-Daoud, "An Improved Implementation of Elliptic Curve Digital Signature by using Sparse Elements," The International Arab Journal of Information Technology, 1(2), pp. 203- 208, 2004.

[11].T. Al-Somani and M.Ibrahim, "Generic-Point Parallel Scalar Multiplication without Precomputation," IEICE Electronics Express, 6(24), pp. 1732-1736, 2009.

[12].B. Ansari and H. Wu, "Parallel Scalar Multiplication for Elliptic Curve Cryptosystems," in Proceedings of International Conference on Communications, Circuits and Systems, Ontario, Canada, vol. 1, pp. 71-73, 2005.

[13].P. Balasubramaniam and E. Karthikeyan, "Fast Simultaneous Scalar Multiplication," Applied Mathematics and Computation, 192(2), pp. 399-404, 2007.

[14].X. Huang, P. Shah, and D, "Sharma, Minimizing Hamming Weight Based on l's Complement of Binary Numbers over GF(2^m)," in Proceedings of the 12th International Conference on Advanced Communication Technology, Piscataway, USA, pp. 1226-1230, 2010.

[15].P. Balasubramaniam, and E. Karthikeyan, "Elliptic Curve Scalar Multiplication Algorithm using Complementary Recoding," Appl. Math. Comput., 190(1), pp.51-56, 2007.

[16].D. A. Booth, "A Signed Binary Multiplication Technique," The Quarterly Journal of Mechanics and Applied Mathematics, 4(2), pp.236–240, 1951.

[17].J.A. Solinas, "Efficient arithmetic on Koblitz curves. Designs, Codes and Cryptography," 19(2–3), pp.195-249, 2000.

[18]. K. Koyama, Y. Tsuruoka, "Speeding up Elliptic Cryptosystems by using a Signed Binary Window Method," In Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag: California, USA, pp–345; .357 1993

[19].I.F. Blake, G. Seroussi, NP. Smart, "Elliptic Curves in Cryptography," Cambridge University Press: Cambridge, UK, 1999.

[20]..K. Okeya, K. Samoa, C. Spahn, and T. Takagi, "Signed Binary Representations Revisited," in Proceeding of Annual International Cryptology Conference Advances in Cryptology Crypto, Santa Barbara, USA, pp. 123-139, 2004.

[21].H. K. Pathak and Manju Sanghi, "Speeding up Computation of Scalar Multiplication in Elliptic Curve Cryptosystem," International Journal on Computer Science and Engineering, 2(4), pp.1024–1028, 2010.

[22].H. Almimi*, A. Samsudin and S. Jahani, "Elliptic-curve scalar multiplication algorithm using ZOT structure," SECURITY AND COMMUNICATION NETWORKS. Vol. 8:pp.1141–1154, 2015.

[23].N. F. H. AL-SAFFAR, "Speeding up the Elliptic Curve Scalar Multiplication Using the Window- Non Adjacent Form," Malaysian Journal of Mathematical Sciences,9(1):pp. 91-110, 2015.

[24].A. Rezai and P. Keshavarzi, "CCS Representation: A New Non-Adjacent Form and its Application in ECC," Journal of Basic and Applied Scientific Research, Vol. **2(5)**, pp. 4577-4586,2012.

[25].V. Suppakitpaisarn and H. Imai, "Optimal Elliptic Curve Scalar Multiplication using Double Base Chain," International Journal of Digital Information and Wireless Communications, Vol. 2(1), pp. 115-134, 2012.

[26].N. Shylashree and V. Sridhar "Efficient Implementation of Scalar Multiplication for Elliptic Curve Cryptography using Ancient Indian Vedic Mathematics over GF (p)," International Journal of Computer Applications, Vol. 49 :pp. 46-50, 2012.

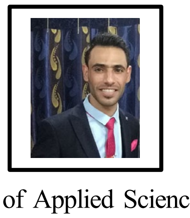
[27].S. William, "Cryptography and Network Security: Principles and Practice," Pearson-Prentice Hall: NJ, USA, 2006.

[28].H. Darrel, JM. Alfred, V. Scott, "Guide to Elliptic Curve Cryptography," Vol. 332. Springer-Verlag: New York, 2003.

[29].M. Stalling, "Cryptography and Network Security," Prentice Hill, USA, 2011

N. M. Ghanim Al-Saidi is a professor in the Department of Applied Sciences, University of Technology-Baghdad-Iraq. She completed her Bachelor of Science and Master of Science degrees in applied mathematics, from Department of Applied Sciences-University of Technology, Baghdad, Iraq, in 1989, and 1995, respectively. She received her Ph.D. degree in mathematics and computer application sciences from Al-Nahrien University, Baghdad, Iraq 2003. She joined the Institute for Mathematical Research (INSPEM), University Putra Malaysia (UPM) as a post doctorate researcher from 2008-2010 with the research project "fractals in Cryptography".

In 1989 she joined the Department of Applied Sciences, at the University of Technology, as an academic staff member. Prof.Dr. Nadia is the author of numerous technical papers since 1994, her research interests include: Cryptography, Fractal geometry, Chaos theory, Graph theory.



M. A. Magamiss is an MSC. researcher in the Department of Applied Sciences, University of Technology-Baghdad-Iraq. He completed his Bachelor of Science and Master of Science degrees in applied mathematics, from Department of Applied Sciences-University of Technology, Baghdad, Iraq, in 2014 and 2016, respectively. His research interests are Abstract Algebra, Cryptography and Mobile security.