Energy efficient elliptic curve point multiplication for WSN applications

Conference Paper · February 2013

DOI: 10.1109/NCC.2013.6488031

3 authors, including:

Ravi Kishore Kodali
National Institute of Technology, Warangal
115 PUBLICATIONS **2,455** CITATIONS

SEE PROFILE

Kashyapkumar Patel
Laurentian University
5 PUBLICATIONS **16** CITATIONS

SEE PROFILE

Energy Efficient Elliptic Curve Point Multiplication for WSN Applications

Ravi Kishore Kodali
Department of E and C E
National Institute of Technology
Warangal, 506004, India

Kashyapkumar H. Patel
Department of E and C E
National Institute of Technology
Warangal, 506004, India

Prof. Narasimha Sarma, NVS
Department of E and C E
National Institute of Technology
Warangal, 506004, India

***Abstract*—Wireless Sensor Networks (WSN's) are resource constrained networks, demanding energy efficient cryptographic algorithms in order to extend security to them. When compared with the popular RSA algorithm, Elliptic Curve Cryptography (ECC) offers similar level of security with smaller key size requirements. An efficient implementation of ECC heavily relies on the scalar multiplication operation. An efficient method for the elliptic curve point multiplication operation is proposed and its resource requirements are compared with binary and non-adjacent form (NAF) techniques. While comparing these, addition and doubling operations are considered. The proposed scalar multiplication technique makes use of both affine and projective coordinate systems while carrying out addition and doubling operations. The number of computations required to perform these operations is compared and an appropriate coordinate system is chosen for each of the operations.**

Keywords– ECC, WSN, Binary method, NAF, w-NAF, coordinate system, Jacobian, Projective

I. INTRODUCTION

Wireless sensor networks (WSN's) have been finding their applications in various diversified areas. In certain WSN applications, the data being exchanged among the nodes within WSN and the nodes and the base station needs to be done securely. For this purpose, symmetrc key cryptographic (SKC) algorithms are being widely used. If the key stored in any of the nodes is compromised, then the attacker can capture the data being exchanged. Alternately, asymetric key or public key cryptographic (PKC) techniques could be used to improve the security in a WSN. Elliptic Curve Cryptography (ECC) [1] is gaining wide acceptance as an alternative to the RSA. As compared to the RSA, the ECC requires smaller key length to provide similar security level [2]. The main attraction of ECC is the hard elliptic curve discrete logarithm problem (ECDLP), which takes a fully exponential time, whereas the RSA takes a sub-exponential time. While RSA requires a 1024− bit key for guaranteeing adequate security, ECC requires only a 160− bit key to provide the same level of security [3], [4].

ECC implementation involves point generation, encoding, and decoding phases. All the phases make use of the point multiplication operation, which in turn uses point addition and doubling sub-operations. Hence, the efficiency of an ECC implementation depends on the algorithm used to carry out the point multiplication [5].

978-1-4673-5952-8/13/$31.00 © 2013 IEEE

The basic form of an elliptic curve over a finite field (F) is given by the Weierstrass equation (1). [6]

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \qquad (1)$$

where $a_1, a_2, a_3, a_4, a_6 \in F$.
An elliptic curve over a Galois field with p elements, $GF(F_p)$, where p is prime and $p > 3$ satisfying the equation (2).

$$E_p(a, b) : y^2 = x^3 + ax + b, \qquad (2)$$

where a and b are constants satisfying $4a^3 + 27b^2 \neq 0$.
Here the elements of the finite field are integers between 0 and $(p - 1)$. All the operations such as addition, subtraction, division and multiplication involve integers between 0 and $(p - 1)$. In order for the cryptosystem to have higher security level, the prime number, p, should be sufficiently large.

II. BACKGROUND MATHEMATICS

Given a scalar K and a point P on an elliptic point, scalar multiplication is defined as KP, which is simply addition of point P with itself K times $\underbrace{(P + P + \cdots + P)}_{\text{K times}}$, so a total of $(K - 1)$ additions are required. The traditional double and add algorithm for point multiplication requires repeated application of the point addition and doubling operations given by the equations (3) and (4), which use the affine coordinate system.

Let $P(x_1, y_1)$, $Q(x_2, y_2)$ and $R(x_3, y_3)$ be points over an elliptic curve $E_p(a, b)$. The point addition ($R = P + Q$) is given by the equation (3)

$$\boxed{x_3 = \Delta^2 - x_1 - x_2 \;\; and \;\; y_3 = \Delta(x_1 - x_3) - y_1,} \qquad (3)$$

where $\Delta = \frac{y_2 - y_1}{x_2 - x_1}$
and point doubling ($R = 2P$) is given by the equation (4)

$$\boxed{x_3 = \Delta^2 - 2x_1 \;\; and \;\; y_3 = \Delta(x_1 - x_3) - y_1,} \qquad (4)$$

where $\Delta = \frac{3x_1^2 + a}{2y_1}$

III. Point Multiplication Techniques

The elliptic curve point addition and doubling computations involve field inversion, addition, subtraction and multiplication operations. Among these operations, the field inversion operation is the most expensive one.

A. Binary point multiplication method

In this method, the scalar multiplier is converted into its binary form. Let P and Q be such that $Q = KP$. K can be represented as

$$\boxed{K = \sum_{i=0}^{m-1} k_i 2^i, \quad k_i \in \{0, 1\},} \tag{5}$$

where $(k_{m-1}, k_{m-2}, ..., k_1, k_0)$ is the binary form of the scalar, K. [7]
This method performs point multiplication by repeatedly applying point addition and doubling operations. For every k_i, point doubling operation is carried out, and in addition, if $k_i = 1$, point addition operation is also performed.

If the length of a scalar in its binary representation is m and n is the number of $1's$ in it, then a maximum of $(m-1)$ doubling operations and $(n-1)$ addition operations are required. Hence, the average computation cost for the binary point multiplication method can be given by the equation (6).

$$\boxed{\mathit{Computation\ cost} = (\frac{m}{2})A + (m)D,} \tag{6}$$

where A and D correspond to addition and doubling operations, respectively.

B. Non-Adjacent Form (NAF) point multiplication method

In the binary point multiplication method, in case the number of $1's$ in the binary form of the scalar is more, then more addition operations are required. In the NAF method, the scalar is represented in its non-adjacent form. For every point, $P(x, y) \in E_p(a, b)$, the negative of the P is represented by $-P = (x, (p - y))$. Thus, subtraction operation is similar to addition operation. The signed bit representation method of the scalar, K, is used as an alternative to the binary form point multiplication method. Each +ve scalar, K, has a unique canonical $\{-1, 0, 1\}$ representation given by the equation (7) [8]. This representation takes a minimum number of non-zero elements, which are non adjacent. In this method, the number of addition operations is reduced.

$$\boxed{K = \sum_{i=0}^{l-1} k_i 2^i, \quad k_i \in \{-1, 0, 1\}} \tag{7}$$

The NAF of a scalar integer is unique and is denoted by NAF(K), which has fewer non-zero digits than any other signed digit representation of K. The NAF(K) requires at most one extra digit when compared with its corresponding binary form. The average density of non-zero digits among all NAF's of length l is about $\frac{l}{3}$ [8].
In this NAF point multiplication method, depending on the present digit value $\in \{-1, 0, 1\}$, one of the following operations is performed:

$$\boxed{\begin{array}{r} -1 : R \leftarrow 2R - P \\ 0 : R \leftarrow 2R \\ 1 : R \leftarrow 2R + P \end{array}}$$

Example: For $K = 63$
Binary representation: 1 1 1 1 1 1
and NAF representation: 1 0 0 0 0 0 -1
In the NAF method, the computation would be as follows: $63P = 2(2(2(2(2(2P))))) - P$
In this example, using the equation (7), it can be observed that the number of point addition operations is reduced by 4 while the number of point doubling operations is increased by 1 when compared with the same example in binary form method. The average computation cost incurred in this NAF method in terms of point addition and doubling operations is expressed by the equation (8).

$$\boxed{\mathit{Computation\ cost} = (\frac{m}{3})A + (m)D,} \tag{8}$$

where A and D correspond to addition and doubling operations, respectively.

C. Windowed-NAF

In the binary form of point multiplication method, after representing the scalar, K, in binary form $\{0, 1\}$, each bit is examined and if the bit is 0 only the doubling operation is performed, otherwise a doubling and addition operations are performed. In the NAF point multiplication method, after representing the scalar, K, in NAF form $\{-1, 0, 1\}$, each digit is examined and if the digit is 0, only doubling operation is done and if the digit is 1, doubling and addition operations are carried out, other wise, i.e. if the digit is -1, doubling and subtraction operations are performed. An extension of the NAF method, Windowed NAF or width-w NAF method, denoted by $\mathrm{NAF}_w(K)$, a +ve scalar integer is represented in the form

$$\boxed{K = \sum_{i=0}^{(l-1)} k_i 2^i, |k_i| \leq 2^{(w-1)}} \tag{9}$$

where each non-zero coefficient k_i is odd, $k_{(l-1)} \neq 0$. In this representation, at most one of any w consecutive digits can be non-zero. The average density of non-zero digits among all the width-w NAF representations of length l is approximately $\frac{l}{w+1}$. Note that if $w = 1$, the representation results in binary form and if $w = 2$, the representation results in NAF form.

The scalar, K, is represented in its width-w NAF form using the Algorithm 1, under the condition that, for any $u \equiv K$ mod 2^w, where $-2^{(w-1)} \leq u < 2^{(w-1)}$, if K is odd, then r is chosen such that, $r = K$ mod $2^{(w-1)}$ and $K - r$ is divisible by $2^{(w-1)}$ [7].

After representing the scalar in its width-W NAF form, the point multiplication operation is computed as given in the algorithm 2.

Algorithm 1 Computing the width-w NAF of a positive integer

```
INPUT: Window width w, positive integer K
OUTPUT: $\mathrm{NAF}_w(K)$
1) $i \leftarrow 0$
2) While $K \geqslant 1$ do
    a) If $K$ is odd then: $k_i \leftarrow K \bmod 2^w$, $K \leftarrow K - k_i$;
    b) Else: $k_i \leftarrow i + 1$.
3) Return$(k_{i-1}, k_{i-2}, ..., k_1, k_0)$.
```

Algorithm 2 Window NAF method for point multiplication

```
INPUT: Window width $w$, positive integer $k$, $P \in E(F_q)$
OUTPUT:$kP$
1) Compute $P_i = iP$ for $i = \{1, 3, \cdots, 2^{(w-1)} - 1\}$.
2) Q← $\infty$
3) For $i$ from $(l - 1)$ downto 0
    a) Q← 2Q
    b) If $k_i \neq 0$ then:
        i) If $k_i > 0$ then Q← Q + $P_{ki}$
        ii) Else Q← Q - $P_{-ki}$
4) Return(Q)
```

This method requires $(2^{(w-2)} - 1)$ pre-computations to be carried out before hand and these values to be stored in the memory. For example, if $w = 5$, then the number of pre-computations is $(2^{(5-2)} - 1) = 7$ and the required pre-computations are: $3P$, $5P$, $7P$, $9P$, $11P$, $13P$ and $15P$.
The average computation cost incurred in this method in terms of point additions and doublings is given by (10). [7]

$$\boxed{Computation\ cost = \underbrace{[1D + (2^{(w-2)} - 1)A]}_{\text{Precomputation overhead}} + [\frac{m}{(w+1)}A + mD],} \tag{10}$$

where A and D correspond to addition and doubling operations, respectively.

IV. Projective coordinate system for point addition and doubling

As discussed in the previous section, point doubling and addition operations are performed at every stage and hence it is imperative to make use of an efficient method to carry out these operations. The equations (3) and (4) for point addition and doubling operations using affine coordinate system require inversion operation, which is very compute-intensive. By mapping the elliptic curve points in affine coordinate system to the corresponding points in projective coordinate system, the need for the inversion operation can be eliminated.

A. Overview

For a prime field F_p, let $p*$ be a subset of non-zero elements $\in \{F_p\}$, λ be any element $\in p*$ and +ve integers be s and t, the projective coordinates, (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) can be defined as an equivalence relation within the set $p^3/(0, 0, 0)$ of non-zero triples over F_p by
$(X_1, Y_1, Z_1) \sim (X_2, Y_2, Z_2)$
where $X_1 = \lambda^s X_2, Y_1 = \lambda^t Y_2, Z_1 = \lambda Z_2$. [9]
This relation can also be expressed as

$$(X : Y : Z) = (\lambda^s X, \lambda^t Y, \lambda Z)$$

$(X : Y : Z)$ is called the projective point, and (X, Y, Z) is called the representative of $(X : Y : Z)$. if $Z \neq 0$ then $(\frac{X}{Z^s}, \frac{Y}{Z^t}, 1)$ is a representative of projective point $(X : Y : Z)$ and there is 1-1 correspondence between the projective point and affine point and their set of points respectively is $P(p)^* = (X : Y : Z) : X, Y, Z \in p, Z \neq 0$ and $A(p) = (x, y) : x, y \in p$. If $Z = 0$ then there is not exit any affine point for projective point.

The projective form of the Weierstrass equation, defined over p, is obtained by substituting in the equation (1):
$x = \frac{X}{Z^s}$ and $y = \frac{Y}{Z^t}$.

B. Standard projective system and mixed coordinate system

If $s = 1$ and $t = 1$, the projective coordinate system is called as the standard projective coordinate system and the corresponding elliptic curve equation under this coordinate system is given by

$$Y^2 Z = X^3 + aXZ^2 + b \tag{11}$$

If the standard projective coordinate point (X, Y, Z), $Z \neq 0$, the corresponding affine coordinate system point is $(\frac{X}{Z}, \frac{Y}{Z})$.
Example
For an affine point $P(11, 3)$ in $GF(F_{17})$ and $Z = 2$, its standard projective point is $(X, Y, Z) = (22, 6, 2) \equiv (5, 6, 2)_{17}$. The point at infinity ∞ is given by $(0 : 1 : 0)$ and the corresponding negative point of $(X : Y : Z)$ is given by $(X : -Y : Z)$.

1) Point addition in standard projective coordinate system: Consider $P(X_1 : Y_1 : Z_1)$, $Q(X_2 : Y_2 : Z_2)$ and $R(X_3 : Y_3 : Z_3)$ in the standard projective coordinate system.
$R = P + Q$ over $E_p(a, b)$ is given by [10]:

$$\boxed{\begin{aligned} X_3 &= BC, && \text{(12a)} \\ Y_3 &= A(B^2 X_1 Z_2 - A) - B^3 Y_1 Z_2, && \text{(12b)} \\ Z_3 &= B^3 Z_1 Z_2, && \text{(12c)} \\ &\text{where } A = Y_2 Z_1 - Y_1 Z_2,\ B = X_2 Z_1 - X_1 Z_2, \\ &C = A^2 Z_1 Z_2 - B^3 - 2B^2 Y_1 Z_2 \end{aligned}}$$

Total computation cost in terms of multiplications (M) and squarings (S) = $12M + 2S$.

2) Point doubling in standard projective coordinate system: Given a point $P = (X_1 : Y_1 : Z_1)$ in the standard projective coordinate system over $E_p(a, b) : y^2 = x^3 + ax + b$, $2P = (X_3 : Y_3 : Z_3)$ is given by [10]:

$$X_3 = 2BD, \tag{13a}$$

$$Y_3 = A(4C - D) - 8B^2 Y_1^2, \tag{13b}$$

$$Z_3 = 8B^3, \tag{13c}$$

where $A = aZ_1^2 + 3X_1^2$, $B = Y_1 Z_1$, $C = X_1 Y_1 B$, $D = A^2 - 8C$
Total computation cost in terms of multiplications (M) and squarings (S) = $7M + 5S$.

C. Jacobian coordinate system

Consider $s = 2$ and $t = 3$, the projective coordinate system is called Jacobian coordinate system and the elliptic curve equation under this Jacobian system is given by

$$Y^2 = X^3 + aXZ^4 + bZ^6 \tag{14}$$

If the Jacobian coordinate point (X, Y, Z), $Z \neq 0$ and its corresponding affine coordinate point is $(\frac{X}{Z^2}, \frac{Y}{Z^3})$. The point at infinity ∞ is given by $(1 : 1 : 0)$ and the corresponding negative point of $(X : Y : Z)$ is given by $(X : -Y : Z)$. In Jacobian coordinate system, the point doubling for a point, $Q(X_1, Y_1, Z_1)$ is given by [7]:

$$X_3 = (3X_1^2 + aZ_1^4)^2 - 8X_1Y_1^2 \tag{15a}$$

$$Y_3 = (3X_1^2 + aZ_1^4)(4X_1Y_1^2 - X_3) - 8Y_1^4 \tag{15b}$$

$$Z_3 = 2Y_1Z_1 \tag{15c}$$

Total computation cost in terms of multiplications (M) and squarings (S) = $4M + 6S$.

V. Proposed method

The computation cost for the first term of the r.h.s., i.e. $(3X_1^2 + aZ_1^4)$ of the equation (15a) = 1M+2S. By substituting $a = -3$ in $(3X_1^2 + aZ_1^4)$ results in:
$3X_1^2 + aZ_1^4 = 3(X_1 + Z_1^2)(X_1 - Z_1^2)$
To do this, the computation cost = $1M + 1S$. As mentioned earlier, the point doubling operation is performed during each iteration of the point multiplication operation and hence this method drastically reduces overall computation cost. For the given point, $P = (X_1 : Y_1 : Z_1)$, $2P = (X_3 : Y_3 : Z_3)$ is given by:

$$X_3 = A^2 - 2D, \tag{16a}$$

$$Y_3 = (D - X_3)A - C^2/2, \tag{16b}$$

$$Z_3 = BZ_1, \tag{16c}$$

where $A = 3(X_1 - Z_1^2)(X_1 + Z_1^2)$, $B = 2Y_1$, $C = B^2$, $D = CX_1$

Algorithm 3 Point doubling ($y^2 = x^3 + ax + b$, using Jacobian coordinate for a=-3)

```
INPUT: $P = (X_1 : Y_1 : Z_1)$ in Jacobian coordinates over $E_p(-3, b) : y^2 = x^3 - 3x + b$
OUTPUT: $2P = (X_3 : Y_3 : Z_3)$ in Jacobian coordinates.
  if $P = \infty$ then Return($P$)
  $A \leftarrow 3(X_1 - Z_1^2)(X_1 + Z_1^2)$
  $B \leftarrow 2Y_1$
  $Z_3 \leftarrow BZ_1$
  $C \leftarrow B^2$
  $D \leftarrow CX_1$
  $X_3 \leftarrow A^2 - 2D$
  $Y_3 \leftarrow (D - X_3)A - C^2/2$
  Return $(X_3 : Y_3 : Z_3)$
```

Total computation cost in terms of multiplications (M) and squarings (S) = $4M + 4S$.

A. Point addition using mixed coordinate system

It is observed that the doubling operation can be performed efficiently making use of Jacobian coordinate system.Whereas, in affine coordinate system, the addition operation requires 1I+2M+1S and in Jacobian coordinate system the addition operation requires 0I+12M+4S, where I represents inversion operation. This work makes use of both the Jacobian and affine coordinate systems,called mixed coordinate system to perform point addition operation. One of the two points is represented in the Jacobian coordinate system while the other is represented in the affine coordinate system. The resulting point after carrying out the point addition operation,is considered to be in the Jacobian coordinate system. Therefore, for a point $P(X_1, Y_1, Z_1)$ in Jacobian coordinate system and $Q(x, y) \equiv Q(X_2, Y_2, Z_2, 1)$ in affine coordinates, where $x_2 = X_2$, $y_2 = Y_2$ and $Z_2 = 1$, $R = P + Q$ is given by [7]

$$X_3 = (Y_2Z_1^3 - Y_1)^2 - (X_2Z_1 - X_1)^2(X_1 + X_2Z_1^2) \tag{17a}$$

$$Y_3 = (Y_2Z_1^3 - Y_1)(X_1(X_2Z_1^2) - X_3) - Y_1(X_2Z_1^2 - X_1)^3 \tag{17b}$$

$$Z_3 = (X_2Z_1^2 - X_1)Z_1 \tag{17c}$$

The point addition can be efficiently performed in mixed coordinate system for a specific elliptic curve, $E_p(-3, b)$, as illustrated in algorithm 4. Total computation cost for point addition in terms of multiplications (M) and squarings (S) = $8M + 3S$.

Algorithm 4 Point addition ($y^2 = x^3 + ax + b$, affine-Jacobian(mixed) coordinates)

```
$P = (X_1 : Y_1 : Z_1)$ in Jacobian coordinate, $Q = (x_2, y_2)$ in affine coordinates on $E(F_p) : y^2 = x^3 + ax + b$.
OUTPUT: $P + Q = (X_3 : Y_3 : Z_3)$ in Jacobian coordinates.
  If $Q = \infty$ then Return$(X_1 : Y_1 : Z_1)$.
  If $P = \infty$ then Return$(x_2 : y_2 : 1)$.
  $A \leftarrow Z_1^2$.
  $B \leftarrow Z_1A$
  $C \leftarrow X_2A$
  $D \leftarrow Y_2B$
  $E \leftarrow C - X_1$
  $F \leftarrow D - Y_1$
    if $E = 0$ then
      if $F = 0$ then use algorithm 3
        to compute $(X_3 : Y_3 : Z_3) = 2(x_2, y_2, 1)$
        and Return$(X_3 : Y_3 : Z_3)$
      else Return (infinite)
  $G \leftarrow E^2$
  $H \leftarrow GE$
  $I \leftarrow X_1G$
  $X_3 \leftarrow F^2 - (H + 2I)$
  $Y_3 \leftarrow F(I - X_3) - Y_1H$
  $Z_3 \leftarrow Z_1E$
  Return $(X_3 : Y_3 : Z_3)$
```

VI. Comparison

A comparison of various scalar point multiplication methods, binary form, Non-Adjacent Form (NAF) and windowed-NAF (w-NAF) is made. Based on the equations (6), (8) and (10), an efficient method for scaler multiplication is proposed. In all these methods, the number of doubling operations required remains to be the same, however the number of point addition operations and the number of pre-computations depend on the window width. The comparison providing the number of additions and pre- computations required for a 160-bit key is given in the Fig 1.

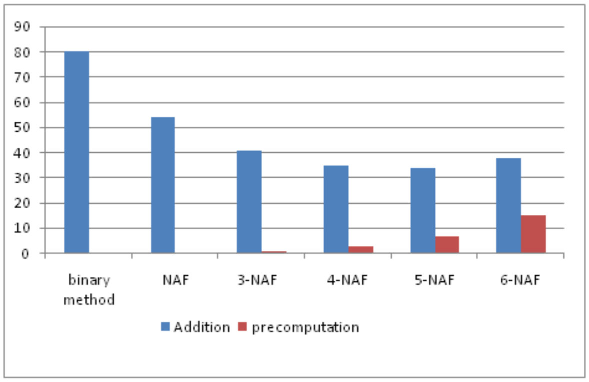

Fig. 1. Comparison Graph

This paper has also discussed various coordinate systems the affine coordinate system, the standard projective coordinate system, the Jacobian coordinate system and mixed coordinate (Jacobian + affine) system to carry out point addition and point doubling operations. A comparison of these coordinate systems, in terms of the total number of required squaring and multiplication operations, is made while carrying out point addition operation using 160-bit key. It can be observed from the Fig 2, point addition operation can be carried out using mixed coordinate system.

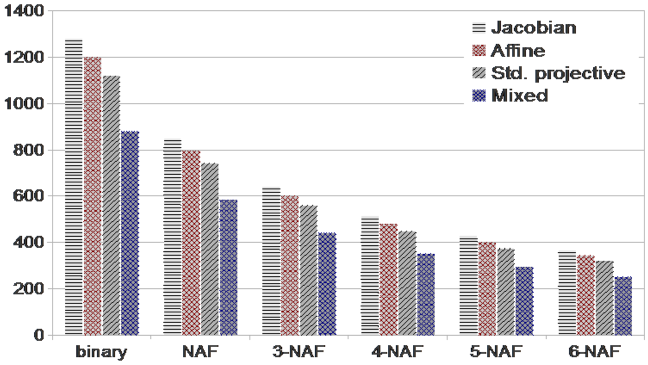

Fig. 2. Point addition Co-ordinate systems Comparison

The computational cost comparison for these schemes is given in Table I. If key length is l, then in Jacobian coordinates, a total of $6l$ doublings are required whereas using a specific value for the elliptic curve parameter $a = -3$, a total of $4l$ doublings are required, which is a 33% reduction in the doubling cost.

TABLE I
Comparison of different coordinate systems

(I - inversion, M - multiplication and S - squaring)

Coordinate System		Doubling			Addition		
		I	M	S	I	M	S
Affine		1	2	2	1	2	1
Projective	$a \in p$	0	7	5		12	2
	$a = -3$	0	7	3	0	12	2
Jacobian	$a \in p$	0	4	6	0	12	4
	$a = -3$	0	4	4	0	12	4
Mixed		-	-	-	0	8	3

VII. Conclusions

ECC can be used in WSN applications as a viable alternative to the RSA in order to provide security. The efficiency of the ECC mainly depends on the point multiplication operation. This work proposes a computationally efficient technique for the point multiplication operation by choosing appropriate values for window width w and $a(-3)$ and giving the result for 160-bit keys. As can be noticed from fig. 1, the windowed NAF(4-NAF) method gives the best result for point multiplication. Additionally, as can be observed from Table I, point doubling operation is best performed using the Jacobian coordinate system, whereas point addition operation is best performed using the mixed coordinate system for the specific elliptic curve $E_P(-3, b)$ over a prime field. By making use of these, while implementing point multiplication, it is possible to improve the energy efficiency of the ECC to be used in WSN applications.

References

[1] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[2] M. Amara and A. Siad, "Elliptic curve cryptography and its applications," in *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*. IEEE, 2011, pp. 247–250.

[3] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE, 2005, pp. 324–328.

[4] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," *Cryptographic Hardware and Embedded Systems-CHES 2004*, pp. 925–943, 2004.

[5] I. Jebrila, R. Sallehb, and A. Mc, "Efficient algorithm in projective coordinates for ecc over."

[6] W. Wei-hong, L. Yu-bing, and C. Tie-ming, "The study and application of elliptic curve cryptography library on wireless sensor network," in *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*. IEEE, 2008, pp. 785–788.

[7] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer, 2004.

[8] E. Karthikeyan and P. Balasubramaniam, "Improved elliptic curve scalar multiplication algorithm," in *Information and Automation, 2006. ICIA 2006. International Conference on*. IEEE, 2006, pp. 254–257.

[9] A. ÖZCAN, "Performance analysis of elliptic curve multiplication algorithms for elliptic curve cryptography," Ph.D. dissertation, MIDDLE EAST TECHNICAL UNIVERSITY, 2006.

[10] A. Gutub and S. Arabia, "Remodeling of elliptic curve cryptography scalar multiplication architecture using parallel jacobian coordinate system," *International Journal of Computer Science and Security (IJCSS)*, vol. 4, no. 4, p. 409, 2010.