Scalar Multiplication Algorithms for Wireless Sensor Network

Syed Hamid Hasan, Anser Ghazzaal Ali Alquraishee
Information Security Research Group
Department of Information Systems
Faculty of Computing and Information Technology
King Abdulaziz University,Jeddah-21589, PO Box- 80221
Kingdom of Saudi Arabia
shh786@hotmail.com

Abstract

Wireless Network Systems are usually deployed in hostile environments where they encountered a wide variety of malicious attacks. A wireless sensor network (WSN) is a wireless network composed of a large number of sensor nodes. In WSNs the scarcest resource is energy which includes low power, less storage space, low computation ability and short communication range. For this reason, algorithmic research in WSN mostly focuses on the study and design of energy aware algorithms for data computation. This problem becomes harder in the case of security, as most of the security algorithms are quite heavy. Although symmetric cryptography may be one of the approaches to solve the problem due to their small computation requirement. In order to further reduce the computational cost of any protocol for WSN, Elliptic curve cryptography (ECC) has been attractive to the researcher due to its smaller key size and its high strength of security. Scalar multiplication is most important cryptographic operation in elliptical curve cryptography. This paper proposes the efficient scalar multiplication algorithms for WSNs which are resistant of side channel attack.

Keywords: *Scalar multiplication, side channel attack, ECDLP*

1 Introduction

Sensor network devices are extremely limited resources in terms of computing, communication, memory, and battery and all the Elliptic Curve Cryptography implementation choices must be made with this aspect in mind. Certainly, the optimal choices for sensor network implementation will be quite different than for a workstation application. In many cases, a single best set of choices is almost impossible to decide on. A more realistic scenario would assume different Elliptic Curve Cryptography parameters for various classes of WSN applications [3].

Elliptic curve cryptography (ECC) was first proposed by Miller and Koblitz in 1985. In virtue of no sub-exponential algorithms known for the elliptic-curve discrete logarithm problem (ECDLP), ECC provides equivalent security strengths with shorter keys compared to other public-key cryptosystems. Due to this advantage, ECC has acquired wide attention in research community as well as industry. However ECC is found to be vulnerable to simple power analysis (SPA) attacks on wireless sensor nodes as other public-key cryptosystems. SPA attackers examine the power traces of cryptographic computations and distinguish the power consumption caused by the secret key. The core operation of ECC is binary scalar multiplication, which consists of a point doubling operation if the key bit is '0', or of a point doubling followed by point addition operation if the key bit is '1'. This is a serious threat for WSNs [3]. Thus, implementers need algorithms that are not only efficient, but also side channel attacks(SCA)-resistant for WSNs. The main target for side channel attacks against ECC implementation on WSNs is the algorithm used for scalar multiplication on the elliptic curve [12] [6]. We can improve several elliptic curve multiplication algorithms secure against side channel attacks (SCA). While some efficient SCA resistant algorithms were developed that apply only to special classes of curves, we are interested in algorithms that are suitable for general elliptic curves and can be applied to the recommended curves found in various standards [8] [5] [9].

1.1 Resources Constraints in WSNs

A wireless sensor network has many resource constraints. The MICA2 mote consists of an 8 bit ATMega 128L macro-controller working on 7.3 MHz. As a result nodes of WSN have limited computational power. Normally, radio transceiver of MICA motes can achieve maximum data rate of 250 Kbits/sec which puts a limitation on the communication resources [3]. The flash memory which is available on the MICA mote is only 512 Kbyte. Apart from these the battery which is available on the board is of 3.3.V with 2A-Hr capacity. Due to the above boundaries the current state of art protocols and algorithms are expensive for sensor networks due to their high communication overheads.

2 Background

This section describes about Elliptic Curve Arithmetics, elementary concepts of Elliptic Curves and Elliptic Curve Discrete Logarithm Problem.

2.1 Elementary concepts of Elliptic Curve

An elliptic curve E over a field K is defined by an equation of the form

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$
Δ is called discriminant of E and is defined as follows

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

$$d_2 = a_1^2 + 4a_2$$

$$d_4 = 2a_4 + a_1a_3$$

$$d_6 = a_3^2 + 4a_6$$

$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2$$

If L is any extension field of K , then the set of L rational points on E is

$$E(L) = \{(x, y) \in L \times L : y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6\} \cup \{\mathcal{O}\}$$

where $\mathcal{O}$ is a special point, called the point at infinity . The equation is called Weierstrass equation [4]. If characteristics of $K \neq 2$ or 3 then the admissible change of variables

$$(x, y) \rightarrow (\frac{x - 3a_1^2 - 12a_2}{36}, \frac{y - 3a_1x}{216} - \frac{a_1^3 + 4a_1a_2 - 12a_3}{24})$$

transform E to the curve $y^2 = x^3 + ax + b$ where $a, b \in K$ and $\Delta = -16(4a^3 + 27b^2)$
We will consider the elliptic curve E of the above simplified equation and have no multiple roots i,e $4a^3 + 27b^2 \neq 0$

2.2 Group Law

Let E be an elliptic curve defined over the field K. There is a chord and tangent rule for adding two point in $E(K)$,if $P_1, P_2 \in E(K)$ then $P_1 + P_2 \in E(K)$ denoted as a third point R which is the reflection of the point of intersection of the chord P_1P_2 to the curve for $P_1 \neq P_2$. If $P_1 = P_2$ then the tangent of $E(K)$ at P_1 gives rise to the point $P_1 + P_2$. The double R of P is defined as follows [7]
First draw a tangent line intersect the elliptic curve at a second point.Then R is the reflection of this point about X-axis. $E(K)$ form an Abelian group with addition operation for $E/K : y^2 = x^3 + ax + b$

1. Identity : $P + \mathcal{O} = \mathcal{O} + \mathcal{P} = \mathcal{P}$, for all $P \in E(K)$
2. Negative : if $P(x, y) \in E(K)$ then $(x, y) + (x, -y) = \mathcal{O}$, The point $(x, -y)$ is dented as -P called negative of P.
3. Point addition: Let $P((x_1, y_1), Q(x_2, y_2) \in E(K)$,then $P + Q = R \in E(K)$ and coordinate $(x_3, y_3)$of R is given by $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$ where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
4. Point doubling : Let $P(x_1, y_1) \in E(K)$ where $P \neq -P$ then $2P = (x_3, y_3)$ where $x_3 = (\frac{3x_1^2+a}{2y_1})^2 - 2x_1$ and $y_3 = (\frac{3x_1^2+a}{2y_1})(x_1 - x_3)$- y_1

Definition 1 Group Order *Let E be the elliptic curve defined over the field F_q. The no. of points in $E(F_q)$ denoted as $\#E(F_q)$ is called the order of E over F_q.*

Theorem 1 Hasse's Theorem *It states that number of points $\#E(F_q) = q + 1 - t$,where $t \leq 2\sqrt{q}$ or we can write $(q + 1 - 2\sqrt{q}) \leq t \leq (q + 1 + 2\sqrt{q})$,$[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$ is called Hasse interval.*

Definition 2 Group Structure *Ler E be the elliptic curve defined over F_q, then E_q is isomorphic to $Z_{n_1} \oplus Z_{n_2}$. Where n_1 and n_2 are uniquely determined positive integer such that n_2/n_1 and $n_2/q - 1$.*
$\#E(F_q) = n_1n_2$, If $n_2 = 1$ then $E(F_q)$ is a cyclic group. If $n_2 > 1$ then $E(F_q)$ is said to have rank 2.

Table 1. Cost of Group Operations in ECC for Various Point Representations for Characteristic > 3

Coordinates	Cost(Addition)	Coordinates	Cost(Doubling)
$\mathcal{A}+\mathcal{A}\rightarrow\mathcal{A}$	$1[i]+2[m]+1[s]$	$2\mathcal{A}\rightarrow\mathcal{A}$	$1[i]+2[m]+2[s]$
$\mathcal{P}+\mathcal{P}\rightarrow\mathcal{P}$	$12[m]+2[s]$	$2\mathcal{P}\rightarrow\mathcal{P}$	$7[m]+3[s]$
$\mathcal{J}+\mathcal{J}\rightarrow\mathcal{J}$	$12[m]+4[s]$	$2\mathcal{J}\rightarrow\mathcal{J}$	$6[m]+4[s]$
$\mathcal{C}+\mathcal{C}\rightarrow\mathcal{C}$	$11[m]+3[s]$	$2\mathcal{C}\rightarrow\mathcal{C}$	$5[m]+4[s]$

3 Point Representation and Cost of Group Operations

Point addition and point doubling are two important operations in (H)ECC. Inversion in a finite field is an expensive operation. To avoid these inversions, several point representations have been proposed in literature. The cost of point addition and doubling varies depending upon the representation of the group elements. This section briefs with some point representations commonly used. Let $[i], [m], [s], [a]$ stand for cost of a field element inversion, a multiplication, a squaring and an addition respectively. Field element addition is considered to be a very cheap operation. In binary fields, squaring is also quite cheaper than a multiplication. If the underlying field is represented in normal basis then squaring is almost for free. Inversion is considered to be 8 to 10 times costlier than a multiplication in binary fields. In prime field the *I/M ratio* is even more.

3.1 Elliptic Curves

Point representation in ECC is a well studied area. In the following two sections we describe some of the point representation popularly used in implementations.

3.1.1 Fields of Characteristic > 3

Elliptic curves over fields of characteristic > 3 have equations of the form $y^2 = x^3+ax+b$. For such curves the following point representation methods are mostly used [10].

1. In *Standard Projective Coordinates* the curve has equation of the form $Y^2Z = X^3 + aXZ^2+bZ^3$. The point $(X : Y : Z)$, with $Z \neq 0$ in projective coordinates is the point $(X/Z, Y/Z)$ in affine coordinates. The point at infinity is represented by the point $(0 : 1 : 0)$ and the inverse of $(X : Y : Z)$ is the point $(X : -Y : Z)$.
2. In *Jacobian Projective Coordinates* the curve has equation of the form $Y^2Z = X^3 + aXZ^4+bZ^6$. The point $(X : Y : Z), Z \neq 0$ in Jacobian coordinates correspond to the affine point $(X/Z^2, Y/Z^3)$. The point at infinity is represented by the point $(1 : 1 : 0)$ and the inverse of $(X : Y : Z)$ is the point $(X : -Y : Z)$. Point doubling becomes cheaper in Jacobian coordinates if the curve parameter $a = -3$.
3. In *Chudonovski Jacobian Coordinates*, the Jacobian point $(X : Y : Z)$ is represented as $(X : Y : Z : Z^2 : Z^3)$. Cost of point addition in Chudonovski Jacobian coordinates is the minimum among all representations.

In Table 1, we present the cost of addition and doubling in the coordinate systems described above. In the table we use $\mathcal{A}$, $\mathcal{P}$, $\mathcal{J}$, $\mathcal{C}$ for affine, projective, Jacobian and

Table 2. Cost of Group Operations in ECC for Various Point Representations in Even Characteristics

Coordinates	Cost(Addition)	Coordinates	Cost(Doubling)
$\mathcal{A}+\mathcal{A}\to\mathcal{A}$	$1[i]+2[m]$	$2\mathcal{A}\to\mathcal{A}$	$1[i]+2[m]$
$\mathcal{P}+\mathcal{P}\to\mathcal{P}$	$13[m]$	$2\mathcal{P}\to\mathcal{P}$	$7[m]+3[s]$
$\mathcal{J}+\mathcal{J}\to\mathcal{J}$	$14[m]$	$2\mathcal{J}\to\mathcal{J}$	$5[m]$
$\mathcal{L}+\mathcal{L}\to\mathcal{L}$	$14[m]$	$2\mathcal{L}\to\mathcal{L}$	$4[m]$

Chudnovski Jacobian respectively. By $2\mathcal{A}\to\mathcal{A}$ we mean the doubling formula in which the input is in affine and so is the output. Similarly for addition and other coordinate systems.

3.1.2 Fields of Characteristic 2

We will consider only non-super singular curves. Elliptic curves (non-super singular) over binary fields have equations of the form $y^2+xy=x^3+ax^2+b$. For such curves the following point representation methods are mostly used.

1. In *Standard Projective Coordinates* the curve has equation of the form $Y^2Z+XYZ=X^3+aX^2Z+bZ^3$. The point $(X:Y:Z)$, with $Z\neq 0$ in projective coordinates is the point $(X/Z, Y/Z)$ in affine coordinates. The point at infinity is represented by the point $(0:1:0)$ and the inverse of $(X:Y:Z)$ is the point $(X:X+Y:Z)$.
2. In *Jacobian Projective Coordinates* the curve has equation of the form $Y^2+XYZ=X^3+aX^2Z^2+bZ^6$. The point $(X:Y:Z)$, with $Z\neq 0$ in Jacobian coordinates correspond to the affine point $(X/Z^2, Y/Z^3)$. The point at infinity is represented by the point $(1:1:0)$ and the inverse of $(X:Y:Z)$ is the point $(X:X+Y:Z)$.
3. In *Lopez-Dahab Coordinates*, the point $(X:Y:Z)$, with $Z\neq 0$ represents the affine point $(X/Z, Y/Z^2)$. The equation of the elliptic curve in this representation is $Y^2+XYZ=X^3Z+aX^2Z^2+bZ^4$. The point at infinity is represented by the point $(1:0:0)$ and the inverse of $(X:Y:Z)$ is the point $(X:X+Y:Z)$.

In Table 2 we present the cost of addition and doubling in the coordinate systems over binary fields. In the table we use $\mathcal{A}$, $\mathcal{P}$, $\mathcal{J}$, $\mathcal{L}$ for affine, projective, Jacobian and Lopez-Dahab respectively. The table follows the same notational convention as in last subsection. we have neglected squaring also. That is because in binary fields squaring is a much cheaper operation than multiplication. It has been reported that [14] if one point is in affine and the other is in projective or some other weighted co-ordinate, then point addition becomes relatively cheaper. This operation is called *addition in mixed coordinates or mixed addition.* In (H)ECC, the base point is generally stored in affine coordinates to take advantage of mixed additions.

4 Scalar Multiplication Algorithms for WSNs

In ECC and HECC, computationally the most expensive operation is scalar multiplication. It is also very important from security point of view. The implementation attacks generally target the computation of this operation to break the cryptosystem on WSNs.

Table 3. Left-to-right and Right-to-left Binary Algorithm

Algorithm DBL-AND-ADD (Left-to-right binary method)	**Algorithm DBL-AND-ADD (Right-to-left binary method)**
Input : $X, m \ (m_{k-1}, \cdots m_1, m_0)$	*Input :* $X, m \ (m_{k-1}, \cdots m_1, m_0)$
Output : mX.	*Output :* mX.
1. $E = m_{k-1}X$	1. $E_0 = X, E_1 = 0$
2. for $i = k - 2$ down to 0	2. for $i = 0$ to $k - 1$
3. $\quad E = \text{DBL}(E)$	3. $\quad$ if $m_i = 1$
4. $\quad$ if $m_i = 1$	4. $\qquad E_1 = \text{ADD}(E_0, E_1)$
5. $\qquad E = \text{ADD}(E, X)$	5. $\quad E_0 = \text{DBL}(E_0)$
6. return E	6. return(E_1)

Given a point X and an positive integer m, computation of $m \times X = X + \cdots (m \text{ times}) \cdots + X$ is called the operation of scalar multiplication. The basic algorithms to compute the scalar multiplication are the age old binary algorithms. They are believed to have been known to the Egyptians two thousand years ago. The two versions of DBL-AND-ADD algorithm are defined above. These algorithms invoke two functions ADD and DBL. ADD takes as input two points X_1 and X_2 and returns their sum $X_1 + X_2$. DBL takes as input one point X and computes its double $2X$.

Both the algorithms first convert the scalar multiplier m into binary. Suppose m has a n-bit representation with hamming weight h. Then, mX can be computed by $n - 1$ invocations of DBL and $h - 1$ invocations of ADD. Hence cost of the scalar multiplication is $(n - 1) \times cost(\text{DBL}) + h \times cost(\text{ADD})$. As the average value of h is $n/2$, on the average these algorithms require $(n - 1)$ doubling and $n/2$ additions. As doubling are required more often than additions, attempts are made to reduce complexity of the doubling operation. The scalar multiplication is the dominant operation in (H)ECC. Extensive research has been carried out to compute it efficiently and a lot of results have been reported in literature. To compute the scalar multiplication efficiently there are three main approaches. As is seen in the basic binary algorithms the efficiency is intimately connected to the efficiency of ADD and DBL algorithms. So the first approach is to compute group operations efficiently. The second approach is to use a representation of the scalar such that the number of invocation of group operation is reduced. The third approach is to use more hardware support (like memory for pre-computation) to compute it efficiently. In some proposals these have approaches have been successfully combined to yield very efficient algorithms. The cost of ADD and DBL depend to a large extent on the choice of underlying field and the point representation. Hence the cost of scalar multiplication also depend upon these choices. Based on the underlying field more efficient operations have been proposed. Over binary fields for ECC, using a point halving algorithm instead of DBL has been proved to be very efficient. Over fields of characteristic 3, point tripling has been more efficient. There are proposals for using more fancier algorithms like the ones efficiently computing $2P + Q$, $3P + Q$ etc. instead of ADD and DBL.

The second approach has also been extensively studied and techniques based on this idea have been successfully employed. We will discuss three techniques of this approach, namely NAF, w-NAF and base-ϕ expansion of the scalar, where ϕ is the Frobenius map.

The third approach is to use more hardware resources to compute the scalar multipli-

(Algorithm: Computation of NAF)

```
Input: An integer $m$.
Output: NAF representation $\Sigma d_i 2^i$ of $m$.
1. $i \leftarrow 0$
2. while $m \geq 1$ do
3.     if $m$ is odd $d_i \leftarrow 2 - (m \ mod \ 4)$
4.     else $d_i = 0$
5.     $m \leftarrow m/2, i \leftarrow i + 1$
6. return $(d_0, d_1, \cdots, d_{i-1})$
```

cation efficiently. This includes methods using pre-computations, parallel methods and pipelining. Many algorithms use a pre-computed table of values to compute the scalar multiplication. These algorithms use more memory for efficiency and are ideal when the base point is fixed. Many parallel algorithms for (H)ECC have been proposed. Recently pipelining techniques have been proposed. All these methods require more hardware support than the basic binary algorithms.

4.1 NAF and w-NAF

The efficiency of computation of scalar multiplication depends upon the hamming weight of the scalar multiplier. The lesser the hamming weight the more efficient is the method. Also, for curves over prime fields, the computation of negation of a point is virtually for free. Hence point addition and point subtraction have almost the same cost. Therefore there are proposals for representing the multiplier in signed binary representation with lesser hamming weight. One such representation is Non Adjacent Form (NAF). We formally define it below.

Definition 3 *A representation of m as $\Sigma_i m_i 2^i$ is in Non Adjacent Form if and only if $m_i m_{i+1} = 0$ for all i.*

The following theorem ensures that every integer has a unique NAF representation.

Theorem 2 *Every integer has a unique NAF representation. This representation has the lowest weight among the signed digit representations and its length is at most one bit longer than the binary representation. The average density of NAF is one third [8] [5].*

The following algorithm computes the NAF of a given integer.

Average hamming weight of an integer of n bits in NAF is $n/3$. Hence, the computation of scalar multiplication requires n doubling and $n/3$ additions on the average. The scalar multiplication algorithm using NAF is similar to the left-to-right and right-to-left algorithm described above, except that the addition step now becomes:

if $d_i = \pm 1$, then $Q = Q \pm P$.

The concept of NAF has been generalized to reduce the complexity of the scalar multiplication algorithm further. The general concept is that of w-NAF. It can be defined as follows:

Definition 4 *Let w be a positive integer. The width w NAF (or briefly w-NAF) representation of a positive integer m is an expression $\Sigma_{i=0}^{l-1} d_i 2^i$, where d_i are odd integers in the range $[-2^{w-1}, 2^{w-1} - 1]$, $d_j \neq 0$ and at most one of any w consecutive integers is nonzero.*

(Computation of w-NAF)

```
Input: An integer $m$.
Output: $w$-NAF representation $\Sigma d_i 2^i$ of $m$.
1. $i \leftarrow 0$
2. while $m \geq 1$ do
3.     if $m$ is odd $d_i \leftarrow 2^{w-1} - (m \; mod \; 2^w)$
4.     else $d_i = 0$
5.     $m \leftarrow m/2, i \leftarrow i+1$
6. return $(d_0, d_1, \cdots, d_{i-1})$
```

(Scalar Multiplication using w-NAF)

```
Input: An integer $m = \Sigma_{i=0}^{n-1} d_i 2^i$ in $w$-NAF.
Output: $mP$.
1. precompute $P_i \leftarrow iP$ for odd $i$ in $[-2^{w-1}, 2^{w-1} - 1]$
2. set $Q = P$
3. for $i = n-1$ to 0
3.     $Q = 2Q$
4.     if $d_i > 0$ then $Q = Q + P_{d_i}$
5.     if $d_i < 0$ then $Q = Q - P_{d_i}$
6. return $Q$
```

The w-NAF representation of an integer m satisfies the following properties:

Theorem 3 *The w-NAF representation of an integer in unique.*

2. *The NAF representation is a special case of w-NAF, with $w = 2$.*
3. *The length of the w-NAF representation of an integer is at most one more than the NAF representation.*
4. *The average hamming weight (number of nonzero digits in the representation) of an integer is $n/(w+1)$, where n is the length of representation.*

The following algorithm calculates the w-NAF representation of a given integer m.

The cost of computation of scalar multiplication goes down drastically if some pre-computation is done, particularly when the base point is fixed. If the base point is not fixed the pre-computation can be done online. One pre-computes the points $P_i = iP$ for odd i in the range $1 \leq i \leq 2^{w-1} - 1$. Note that the computation of negation of these points is very cheap(virtually "for free" over prime fields). So P_i for negative i's need not be computed, which also reduces the storage requirement. The following algorithm computes the scalar multiplication [14].

The above algorithm requires 1 doubling and $2^{w-2} - 1$ additions for the pre-computation. The scalar multiplication computation requires n doublings and $n/(w+1)$ additions on the average [4].

4.2 Frobenius Map

Koblitz had suggested the use of Frobenius map to speed up scalar multiplication algorithm. For hyperelliptic curves, it has been shown that the Frobenius map based method can be used over any field of finite characteristic. Let q be a prime power, F_q be the finite

```
(Scalar Multiplication Using Frobenius Map)
Input : integer $m = \sum_{i=0}^{n-1} u_i\phi^i$ and point $X$.
Output : $mX$.
    1. For $0 \le i \le n-1$ and $0 \le j \le A$, compute $X_i$ and $u_{i,j}$;
    2. Set $Y = \sum_{i=0}^{n-1} u_{i,A}X_i$;
    3. For $j = A-1$ down to 0
    4.         $Y = 2Y$; $Y = Y + \sum_{i=0}^{n-1} u_{i,j}X_i$
    5. return $Y$.
```

field of order q and F_{q^n} an extension field of F_q. Let C be the curve of genus g to be used for the cryptosystem and we consider the F_{q^n}-rational points of C. The Frobenius map ϕ : $F_{q^n} \to F_{q^n}$ is an automorphism of F_{q^n} and is defined as $\phi(x) = x^q$. The map is extended to points of an elliptic or hyperelliptic curve over F_{q^n} in the following manner: A point of an elliptic curve is represented using a pair of elements of F_{q^n}; similarly a reduced divisor of a hyperelliptic curve is represented using a tuple of elements of F_{q^n}. An application of the Frobenius map [1] to a point is to actually apply the map individually to the field elements which represent the point. We note that ϕ^n is the identity map on F_{q^n}. If the field F_{q^n} is represented using a normal basis, then the computation of $\phi(x)$ is "for free". It has been observed that in the case $q = 2$, the Frobenius map is $\phi(x) = x^2$ and hence can be computed using a field squaring which is a relatively cheap operation even if polynomial basis representation of elements is used. Let m be an integer, X a point (either a point of an elliptic curve or a reduced divisor of a hyperelliptic curve) and we wish to compute mX. The base-ϕ expansion of m is $\sum_{i=0}^{n-1} u_i\phi^i$, where under reasonable assumptions each u_i is an integer in the range $[-q^g, q^g]$. It is possible to obtain the base-ϕ expansion for each integer m. Next we define the following parameters:

1. $A = \max\lfloor \log_2(|u_i|)\rfloor$.
2. For $i \in \{0, \ldots, n-1\}$ write $|u_i| = \sum_{j=0}^{A} u'_{i,j}2^i$, where $u'_{i,j} \in \{0, 1\}$.
4. For $0 \le i \le n-1$, define $X_0 = X$ and $X_i = \phi^i(X_0) = \phi^i(X)$.

The expression mX can be written as

$$\left.\begin{array}{rcl} mX & = & u_0X_0 + u_1X_1 + \cdots + u_{n-1}X_{n-1} \\ & = & (u_{0,0} + u_{0,1}2 + \cdots + u_{0,A}2^A)X_0 \\ & & +(u_{1,0} + u_{1,1}2 + \cdots + u_{1,A}2^A)X_1 \\ & & +\cdots \\ & & +(u_{n-1,0} + u_{n-1,1}2 + \cdots + u_{n-1,A}2^A)X_{n-1} \end{array}\right\} \quad (2)$$

Theorem 4 *In the above algorithm, the average numbers of additions and doubling needed to compute mX are $n(A+1)/2$ and A respectively.*

5 Elliptic Curve Addition and Multiplication Algorithms

The two operation in scalar multiplication algorithm are *elliptic curve adding* (ECADD) and *elliptic curve doubling* (ECDBL). Both algorithms and their efficiency are described below.

5.1 Addition and Doubling Algorithms and their Efficiency

The curve in the simplified form of *Weierstress Equation* is given by

$$E : y^2 = x^3 + ax + b \tag{3}$$

The coefficient a is arbitrary field element.However many curve recommended by specification such as [NIST,ANSI,SEC2] use $a = -3$ for more efficient ECDBL implementation. The general algorithm ECDBL requires time $4M + 6S + 11A$, where M,S and A denote time needed for multiplication,squaring and adding respectively. This uses 6 auxiliary variables.The optimized algorithm ECDBL for $a = -3$ requires $4M + 4S + 3A$ using 5 auxiliary variables.

5.2 Window Based Methods

Window based methods come under the third category of scalar multiplication algorithms, which require higher amount of computational resources. Window based methods use higher amount of memory as they use a pre-computed table and are more efficient if the base point is fixed. The target of these methods is to minimize the size of the look-up table and maximize the gain in performance. Algorithms using w-NAF are window based methods.

6 Conclusion

This article proposes efficient algorithms for scalar multiplication on Elliptic Curve can be applied on Wireless Sensor Network. These are secured against side Channel Attack. The window size may be a subject of trade off between the available RAM and ROM at that particular instance on sensor node. As NAF method involves modular inversion operation to get the NAF of binary number, the one's complement subtraction can provide a very simple way of recoding integer [9]. Due to physical characteristics of sensor node, the power consumption and time consumption using the secret key can be clearly observed. Thus Side Channel Attack(SCA)are a serious threat against these devices. The main target for Side Channel Attack(SCA) against ECC implementation is the algorithm used for scalar multiplication on elliptic curve .Therefore various elliptic curve multiplication algorithms designed to resist Side Channel Attack and Differential Power Analysis Attack have been proposed.

References

[1] N. Koblitz Elliptic Curves in Cryptography In Journal of Cryptology.

[2] D. Henkerson, A. Menezes, S. Vanstone Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.

[3] J.P. Kar "Non-interactive Deniable Authentication Protocol using generalized ECDSA Signature Scheme" International Journal of Smart Home. Vol.5 (4), pp 39-49, October 2011.

[4] E. Brier and M. Joye Weierstrass Elliptic Curves and Side-Channel Attacks

[5] B. Moller Securing Elliptic curve point against side channel attack

[6] A. Menezes,P.C. van Oorschot, and S.A. Vanstone. Hand book of applied cryptography,C.R.C press,1997

[7] N. Koblitz. A course in Number Theory and Cryptography ,2nd edition Springer-Verlag-1994

[8] C. Kocher, J. Jaffe and B. June . Differential Power Analysis,CRYPTO99

[9] K. Okeya, K. Miyazaki and K. Sakuari . A first scalar method with randomized projective coordinates on a Montgomery form Elliptic curve secure against side channel attacks

[10] E. Brier and M.Joye . Weierstrass elliptic curves and side channel attacks,Springer-2002

[11] K. Koc, David Naccache, and Christof Paar . Cryptograhic Hardware and Embbeded Systems,Springer -Verlag-2001

[12] D. R. Stinson . Cryptographic :theory and practice,CRC Press, Ibc, 1995

[13] V. S. Miller . Use of Elliptic Curves in Cryptography.Advances in Cryptology CRYPT '85

[14] H. Cohen, A. Miyaji, and T. Ono. Efficient Elliptic Curve Exponentiation Using Mixed Co-ordinates, In Asiacrypt'98, LNCS 1514, pp. 51-65, Springer-Verlag, 1998.

Authors

Dr.Syed Hamid Hasan has received his PhD in Computer Science from JMI, India, MSc in Statistics from AMU, India. Also he has completed Post-Graduate Diploma in Computer Science from the same university. Prof. Hamid has worked as a Head of Computer Science department at the AMU, India and was also Head of IT department at the Musana College of Technology, Sultanate of Oman. Dr Hamid is currently working as a Professor at Information Systems department, faculty of Computing and Information Technology and leading the Information Security Research Group, King Abdulaziz University, Kingdom of Saudi Arabia. He is reviewer of various research articles of peer reviewed International Journals and conferences. He is included in the Panel of referees of "The Indian journal of community health", was Chief Coordinator of the National Conference on "Vocationalization" of Computer Education" held on 28-29 September-1996 at A.M.U. Aligarh-India. He is a life Member of Indian Society for Industrial and Applicable Mathematics (ISIAM), Computer Society of India, Fellow National Association of Computer Educators & Trainers (FNACET), India. He has 23 research articles in conferences & journals to his credit. His research interest is e-Security and Cryptography.

Anser Ghazzaal Ali AlQuraishee has received his M.Sc. in Computer Science from University of Punjab Lahore, Pakistan. He has worked as Software Architect and System Analyst. He is well versed in Oracle development Tools and has successfully Implemented suite software projects on Cryptographic Protocols. This includes digital signature using Elliptic Curve Cryptography and key agreement protocols for low processor devices. He has proven abilities for system analysis, design strategies and Development Skills. Managed, analyzed, designed and developed versatile Software Systems for Organizations ranging from mid to large-size. Oracle Certified Professional for Developer 2000 R2, Oracle Developer 6/6*i* and Oracle Database 8*i*. Currently he is pursuing Ph.D at Universiti Teknologi, Malaysia. His current research interests are on development and design of provably secure cryptographic primitives and protocols for Wireless Sensor Networks.