Algorithm based on Booth's Encoding Pattern for Fast Scalar Point Multiplication in ECC for Wireless Sensor Networks

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

Ravi Teja Reddy Levaka

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
May 2011

Algorithm based on Booth's Encoding Pattern for Fast Scalar Point Multiplication in ECC for Wireless Sensor Networks

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

Ravi Teja Reddy Levaka

(Roll- 209CS2089)

Supervisor

Prof. P.M.Khilar

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

May 2011

Dedicated to My Parents

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***Algorithm Based on Booth's Encoding Pattern for Fast Scalar Point Multplication in ECC for Wireless Sensor Networks*** by ***Ravi Teja Reddy Levaka*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Information Security in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: 30 May 2011

Dr. P.M.Khilar
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. P.M.Khilar for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am very much indebted to Prof. Ashok Kumar Turuk, Head-CSE, for his continuous encouragement and support. He is always ready to help with a smile. I am also thankful to all the professors of the department for their support.

I am really thankful to my all friends. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

Ravi Teja Reddy Levaka
ravitejareddylevaka@gmail.com

Abstract

With the rapid increase of small devices and its usage, a better suitable security providing mechanism must be incorported keeping the resource constraints of the devices in mind. Elliptic Curve Cryptography (ECC) serves the best and highly suitable for wireless sensor Networks (WSN) in providing security because of its smaller key size and its high strength of security against Elliptic Curve Discrete Logarithm Problem (ECDLP) than any other public-Key Cryptographic Systems. But there is a scope to reduce key calculation time to meet the potential applications, without compromising in level of security in particular for wireless sensor networks. Scalar Multiplication is the costliest operation among the operations in Elliptic Curve Cryptography which takes 80% of key calculation time on WSN motes. This research proposes an algorithm based on Booth's Encoding Pattern, offering minimal Hamming Weight and significantly reduces the computational cost of scalar multiplication. Simulation results has proved that the Booth's encoded pattern performs better over the existing techniques if there are atleast 46% number of 1's in the key on an average.

Contents

List of Figures

List of Abbreviations

WSN	Wireless Sensor Network
NAF	Non-Adjacent Form
TNAF	T-adic Non-Adjacent Form
ECC	Elliptic Curve Cryptography
RSA	Rivest-Shamir-Adleman
BEP	Booth's Encoding Pattern
IFP	Integer Factorization Problem
DLP	Discrete Logarthm Problem
DSA	Digital Signature Algorithm
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
GF	Galois Field
JSF	Joint Sparse Form
MSB	Most Significant Bit
LSB	Least Significant Bit
SDR	Signed Digit Representaion
SDN	Signed Digit Number
OCS	One's Complement Subtraction

Chapter 1

Introduction

Security goals

Motivation

Thesis Organization

Chapter 1

Introduction

Rapid growth in use of small computer devices with wireless communication capabilities drive the future development of Internet Applications. The number of people using small devices far exceeds than that of Personal Computers. To tap this potential the security issues with applications on these devices have to be addressed.As the resources like battery power, memory, small processors are limited in such devices, the level of security is need to be addressed efficiently. Several approaches to enhance the security are based on cryptographic primitives such as message authentication codes, hash functions, and digital signatures. The limitations of providing a high level of security using these primitives include: the memory needed for generating cryptographic keys; the storage needed for storing the key generating algorithms as well as the keys; the bandwidth necessary to transmit keys; and the CPU to generate keys [1].

Elliptic Curve Cryptography has emerged as an attractive public-key cryptosystem for use in small wireless environments. Compared to the conventional cryptosystems like RSA, ECC offers a higher level of security with small key sizes, resulting in faster computations, lower power consumption, as well as memory and bandwidth savings [2]. For the same key sizes, ECC provides higher security than that of RSA. These properties make ECC very useful on small mobile devices and smartcards which are typically limited in terms of their CPU, power and network connectivity.

1.1 Security goals

The keen observation of a loosely packed communication channel between the communication entities reveals the following fundamental objective of secure communication:

- *Confidentiality*: Data meant to be secret has to be secret for all unauthorized , but only authorized should have a way to access the original communication i.e.; message between two communicating parties A and B should not be readable by E (any third-party entity).
- *Data Integrity*: Data should not be altered by unauthorized means in any way
- *Authentication*: Only the authorized entities are given access to obtain the information.
- *Availability*: The information created and stored needs to be available to authorized entities. Information is useless if it is not made available.

Key forms the most important component in the today cryptography systems. They are numbers randomly choosen from a large pool of numbers in a set.Thus, management of keys plays an important role and includes the following:

- *Key Generation*: It is the process in which a pool of keys are generted.
- *Key Establishment*: It is the most important phase of key management process. It deals how the keys are determined to each and every node in the network.
- *Key Updation*: It is the process by which we can update the keys of all the users after regular time intervals to keep the network secured.
- *Key Revocation*: This process is like renewing the keys once they are known to be compromised.

Based on the usage of the key, Cryptographic algorithms can be divided into two categories.

- *Private Key Cryptography*: Same key is used for encryption and decryption sessions. It is common-shared secret key between the communicating entities. The key is known to be as secret key.
- *Public Key Crytography*: Two types of keys public key and private keys are used in the system. Every party involved in the communication has to have the both of the keys where as public key is known to the world and private key is only known to that party.

As public key cryptography deals with the computation on large numbers,it needs huge computational and communication costs where as Private key cryptography deals with substitution or permutations of the characters , so it is faster in computation involved. But in terminology, both has to go hand−in−hand where each has its own advantages and disadvantages when compared with each other. To be able to use all the aspects of security, both Private−Key and Public−Key Crytography techniques are needed.

The main idea behind Public Key Cryptography is the concept of the Trapdoor One−Way Function.

- *One− Way Function*: It is function satisfying the following two properties.
  - Function $y = f(x)$ can be easily computed.
  - Given y, comptuing Inverse of the function $x = f^{-1}(y)$ is computationally infeasible.
- *Trapdoor One− Way Function*: It is a one−way function along with another property
  - Given y, and a trapdoor(Private Key) k, x can be computed at ease.

There are several criteria that should be considered when selecting a family of public-key cryptography schemes. A few are [3]:

- *Functionality*: The selected scheme has to provide the desired capabilites.
- *Security:* The security need to be provided has to be assured.
- *Performance:* For the security level provided, the protocol has to meet the performance objectives.

RSA, ELGAMAL and ECC are some of the schemes providing all these functionalities expected of Public−Key Cryptography: Key Management, Signatures and Encryption.The fundamental security issue that remains is the hardness of the underlying mathematical problem that is necessary for the security of all protocols in a public-Key family − the integer factorizatin problem for RSA systems, the discrete logarithm problem for DL systems, and Elliptic Discrete Logarthim Problem for EC systems [3].

1.2 Motivation

Wireless sensor networks contain hundreds or thousands of sensor nodes that are resource specific like limitted battery power, lesser memory, lower computation processing speeds etc., where a small task is processed by all the nodes. Security plays a vital role in order to protect each and every sensor node as the information is revealed by intruder on compromise of a single sensor node.So many light-weight architectures have been proposed to accomodate the characteristics of such small devices. So the reduction in computation in the node increases the lifetime of the node, thus increasing the lifetime of the network. Dharmendra sharma [4] has discussed an approach based on one's complement subtraction to reduce the hamming weight in the key by recoding it, thus reducing the computation cost. Our motivation is to reduce the computation cost, further. We have recoded the integer key using Booth's encoding pattern and has got still lesser hamming weight , suceeding in reducing the computation cost and speeding the operation further. The proposed approach doesnt need any memory overheads or precomputations.

1.3 Thesis Organization

In this thesis, we have discussed about Booth's Encoding Pattern of the integer key and how it outperforms the existing recoding shcemes. In chapter-1, we have discussed the introduction of Wireless Sensor Networks and Elliptic Curve Cryptography. The rest of this paper is organized as follows. In Chapter-2, we have discussed about the background of Elliptic Curve Cryptography and its Preliminaries, Elliptic Curve Point Multiplication and some of the existing methods of point multiplication in ECC and its Related Work. Chapter-3 goes through the Simulation and Results. Chapter-4 concludes our thesis.

Chapter 2

Background

Background of Elliptic Curve Cryptography and its Preliminaries

Elliptic Curve Point Multiplication

Recoding of Integer k in PointMultiplication

Background of PointMultiplication

Summary

Chapter 2
Background

This chapter is divided into three parts. In the first part, we have presented the Elliptic Curve Cryptosystem and its Preliminaries. In the second part, we have explained the basics of scalar and point multiplication. In the third part we have discussed about the various existing point multiplication techniques and the related work.

2.1 Background of Elliptic Curve Cryptography and its Preliminaries

Elliptic Curve Cryptography(ECC) has emerged as an attractive public-key cryptosystem for use in small wireless environments. The advantage of Elliptic Curve Cryptography over other public key cryptography techniques such as RSA, Diffie-Hellman is that Key sizes of ECC are lesser and the best known algorithm for solving ECDLP is the hard mathematical problem that takes the fully exponential time [5]. On contrary, the best algorithm for solving RSA and Diffie-Hellman takes sub-exponential time. To keep in short, ECC can be solved only in exponential time and so far there is lack of known sub-exponential attack on ECC.

Elliptic curve cryptography is an approach to public-key cryptography based on the algebraic structures of elliptic curves over finite fields [6]. It was introduced by Vector and Miller independently in eighties [7] [8]. The elliptic curves in cryptography are typically defined over two types of finite fields: prime fields F_p, where p is a large prime number, and binary extension fields F_{2^m}. Although both implementations were done for WSN networks, the more efficient in terms of space and processing is elliptic curves over prime fields F_p. The number of elements in the field is called its order. There exists a finite field of order q if and only if p is of a prime power. If $q = p^m$, where p is prime and a positive integer, then p is called the characteristic and the m extension degree of F_q. When setting up an elliptic curve cryptosystem, there are three basic decisions that need to be made: [3]

- Selection of the underlying finite field F_p.
- Selection of the representation for the elements of F_p.
- Selection of the elliptic curve E over F_p.

2.1.1 Domain Parameters

These parameters for an Elliptic Curve scheme describe the Elliptic Curve E defined over a finite field F_q, a base point P $\epsilon E(f_q)$ and its order n. These parameters are chosen such that the ECDLP is resistant to all the attacks. These Domain parameters are shared by the group of entities involved in the communciation.

Domain Paramters D=(q, FR, S, a, b, P, n, h) are: [3]

- *Field Order q:* prime field F_P or binary field F_{2^m}.
- *Field Representation FR:* Representation used for the elements of F_q
- *Seed S:* if the elliptic curve was randomly generated
- *Coefficients a,b:* defines the equation of the curve
  - $y^2 = x^3 + ax + b$ if the field chosen is Prime Field, where determinant $4a^3 + 27b^3 \neq 0$ should be satisfied.
  - $y^2 + xy = x^3 + ax^2 + b$ if the field chosen is Binary Field, where $b \neq 0$
- *Base Point P:* $P(x_P, y_P)$ on the equation has prime order.
- *Order n:* The least integer when multiplied with the base point P result in point at infinity O.
- *Cofactor h:* $E(F_q)/n$.

Since the group is abelian a point known as Point at Infinity O is also included in the point set of the equation, which serves as additive identity of the group satsifying the abelian property.

The three opertaions Point-Addition, Point-Double and Point-Negation on the points of the Elliptic Curve are define as:

Let P(x_1, y_1) and Q(x_2, y_2) be two different points on the elliptic curve.The operations on the points are defined in the following algorithms.

2.1.2 Point-Addition:

R(x_3,y_3) = P(x_1,y_1) + Q(x_2,y_2) on the curve is defined[Fig 2.1] as:

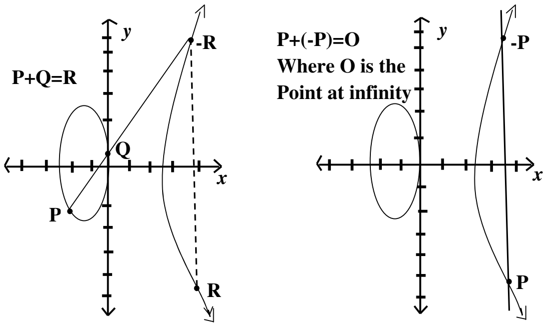


Figure 2.1: Point Addition

Algorithm :

- step 1: check if either point is O,then R = the other point.
- Step 2: If P = Q , use POINT DOUBLE routine.
- Step 3: If $x_1 = x_2$, R = O;
- Step 4: If P $\neq$ Q, R(x_3,y_3) where
  $\lambda = (y_2 - y_1)/(x_2 - x_1)$;
  $x3 = \lambda^2 - x_1 - x_2$;
  $y3 = (x_1 - x_3)\lambda - y1$;

2.1.3 Point-Double:

R(x_3,y_3) = P(x_1,y_1) + p(x_1,y_1) on the curve is defined[Fig 2.2] as:

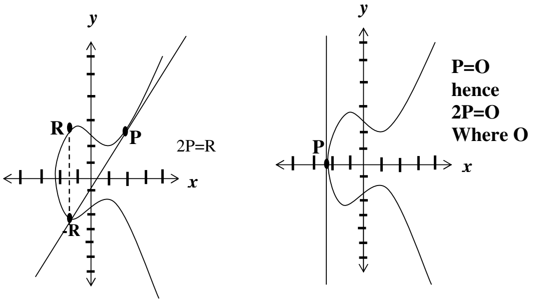


Figure 2.2: Point Double

Algorithm :

- step 1: check if the point is O,then R = O.
- Step 2: If P $\neq$ O , R(x_3, y_3) where
  $\lambda = (3x^2 + a)/2y_1$;
  $x_3 = \lambda^2 - x_1 - x_2$;
  $y_3 = (x_1 - x_3)\lambda - y_1$;

2.1.4 Point-Negation:

Let P(x$_1$,y_1) be a point on the curve. R(x_3,y_3) =-p on the curve is defined as

Algorithm :

- step 1: R(x_3, y_3) = -(x_1, y_1) = (x_1,x_1+y_1);

where the addition, subtraction, multiplication and inverse are the arithmetic scalar operations over the field GF(p).

2.2 Elliptic Curve Point Multiplication

In Elliptic Curve Cryptography, exponentiation operation of RSA and several crypographic algorithms like Elgamal Cryptosystems, is replaced by an operation called Point Multiplication, where its really impracticable to break, easy to use and computation sensitive. ECDLP hardness lies in the operation Q = k.P where P and Q are any arbitrary points on the curve, and is impracticable to deceive k though P,Q are known. This forms the costliest and very important operation in ECC.

2.2.1 Ellipitc Curve Point Generation

In ECC simulating ElGamal Cryptosystems, in Key Generation stage at a communication entity

$$e_2 = d \times e_1 \tag{2.1}$$

is the point multiplication operation where e_1 and e_2 are coordinate points ,forming part of public key. d forms the private key. e_2 is a point generated by multiplying another point e_1, k number of times where k is the randomly selected large number that serves as private key.

2.2.2 Elliptic Curve Diffie-Hellman Key Exchange

This scalar point multiplication is used to obtain the common shared key between two communication entities in the network.

Let k_a and k_b are the private keys of two nodes A and B respectively. Correspondingly, node A and node B calculates the intermediate keys using the scalar point multiplication Q_a=k_a.G and Q_b=k_b.G and exchanges mutually. To obtain the shared common key between the communicating parties

- A calculates $k_a.Q_b$
- B calculates $k_b.Q_a$

- Thus,the shared key is obtained and verified as

$$k_a.k_b.G = k_b.k_a.G \tag{2.2}$$

Theoretically, Scalar Point Multiplication operation k.P is calculated as
Q = k.P = (P+P+...+P) i.e. k times

2.3 Recoding of Integer k in Point Multiplication

Computational cost significantly varies by the recoding of the integer k in the scalar point multiplication operation. The number of point doubling and point additions in scalar multiplication depends on the coding pattern of the integer k. The number of ones and zeros in the binary form, their places and the total number of bits in the integer key k affects the computational cost of the operation. The number of non-zero bits,in the binary form, known as hamming weight determines the point additions, where as the point double operations are determined by the total bit length of the key k [4].

One point addition requires one field inversion and three field multiplications. Squaring is counted as regular multiplication. One point double operation requires one field inversion and four multiplications. And additions can be neglected as multiplication cost is much more than that of addition cost and multiplication with small constant is also neglected [4].

2.4 Background of Point Multiplication

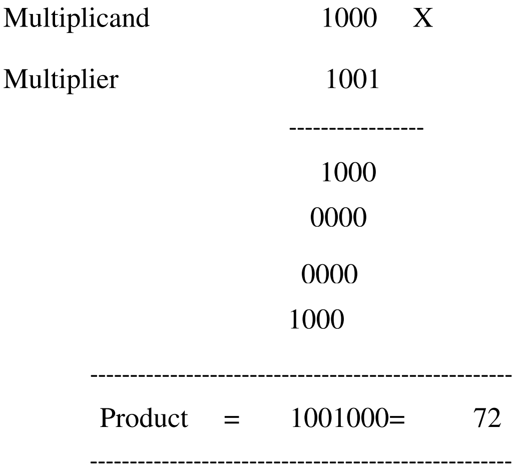
The Q=k.P operation is implemented practically by recoding the integer k in significant format and either a left-to-right or right-to-left scan of k is performed along with one of the following existing methods.In our approach, we strict to left-to-right scan of the integer key i.e.most significant bit to least significant bit when the internal storage architecture is Little Endian format.

2.4.1 Basics of Multiplication

A machine is only capable of storing two bits 0 and 1, denotion of +5V and 0V electric signals. Every multiplication in hardware is done by representing the multiplicand and multiplier in binary form.

2.4.1.1 Multiplication of scalar with a scalar

When both the multiplicand and multiplier are scalars, Shift-and-add multiplication method is used to obtain the product, where shift does the double opertaion of the multiplicand and add does the addition of multiplicand with the multiplier. This method adds the multiplicand P to itself k times, where k denotes the multiplier. To multiply two numbers, the algorithm is to take the bits of the multiplier one at a time from right-to-left or left-to-right, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results.
As an example, consider the multiplication of two unsigned 4-bit numbers, 8 (1000) and 9 (1001) [Fig 2.3].



```
Multiplicand              1000    X

Multiplier                1001
                        ----------------
                          1000
                         0000

                        0000
                       1000
                ------------------------------------------------
                 Product    =    1001000=      72
                ------------------------------------------------
```

Figure 2.3: Multiplication of scalar with a scalar

2.4.1.2 Multiplication of point with a scalar

When a elliptic curve point is multiplied with a scalar integer, point acts as multiplicand and the scalar as multiplier. The product can be obtained in one of the following two ways , where the multiplier has to be processed either from left-to-right or from right-to-left multiplying with the multiplicand for each bit in the coded pattern of scalar integer. The scalar can be any of the two representations [4].

- *Binary-Digit Representaion Method* In this method the integer k is represented in its binary form
  k = $\sum_{i=0}^{l-1} 2^i k_i$, where $k_j \epsilon \{0, 1\}$
  In this method, only two bits 0 and 1 are used to represent the integer.

- *Signed-Digit Representation Method* In this method the integer k is represented in its canonical form
  k = $\sum_{i=0}^{l-1} 2^i k_i$, where $k_j \epsilon \{0, 1, -1\}$
  The subtraction operation is typically of same cost as addition in the elliptic curve group. The negative of point (x, y) is (x,-y) in ECC operations. This leads to scalar multiplication methods based on additionsubtraction chains, helping to reduce the number of curve operations. When integer is represented with the following forms, it is called as binary signed digit representations.

2.4.2 Binary - Add and Double Method

Add and Shift method of integer Arithmetic is similar to that of Add and Double method, as we deal with the point double operations in ECC, shifting the bits in the integer multiplier to obtain the product. This is obtained by repeatedly applying of elliptic curve point add and double operations.Algorithm 1 explains the Add and Double method to compute the product by Right-to-Left Scanning of the scalar integer.

Algorithm 1 Right-to-Left Scan of k

```
Input: $k = (k_{l-1}k_{l-2} \dots k_0)$ where $k_i \epsilon 0, 1$.
Output: kP
Initialize: Q=0
for $i = 1 \rightarrow l - 1$ do
    if $k_i == 1$ then
        Q:=Q+P
    end if
    P:=2P
end for
return Q
```

[Fig 2.4] explains the scalar point multiplication between two integers 10 and 25,where 10 is assumed as scalar multiplier and 25 as point multiplicand based on the Algorithm 1.

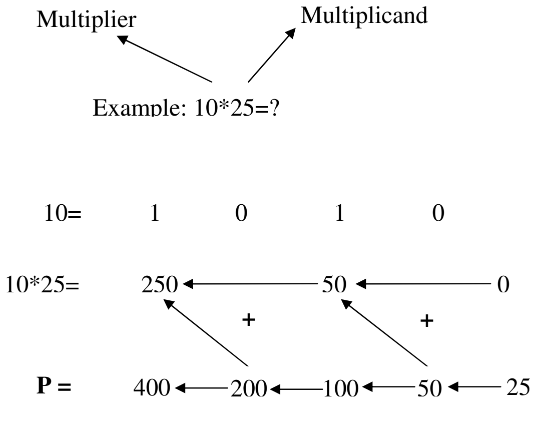


Figure 2.4: Right-to-Left Scan Method

The cost of multiplication in Add and Double method depends on the number of non-zero elements and length of the binary representation of k. If the representation has $k_{l-1} \neq 0$ then this method require $l - 1$ point double operations and

$h-1$ point addition operations, where l is the length of the binary expansion of k and h is the Hamming weight of the k that is the number of non zero elements in expansion of k i.e.
if $k = 723 = (1100010011)_2$, total h-1=4 point additions and $l-1 = 9$ point double operations are required. when point at infinity is added with a point or doubled itself there is no computation involved.

2.4.3 Binary - Double and Add Method

Algorithm 2 Left-to-Right Scan of k

```
Input: $k = (k_{l-1}k_{l-2}\dots k_0)$ where $k_i \epsilon 0, 1$.
Output: kP
Initialize: Q=0
for $i = l-1 \rightarrow 0$ do
    Q:=2Q
    if $k_i == 1$ then
        Q:=Q+P
    end if
end for
return Q
```

The cost of the point multiplication opertaion of Left-to-Right Scan method is often lesser than the Right-to-Left Scan method, since in certain settings of Elliptic Curves the latter method outperforms the former one. And in the former case in every pass of the algorithm, multiplier is updated and multiplicand scalar is not disturbed, where as in the other both are updated in each and every pass.we strict out attention towards Left-to-Right Scan method only as it is more convenient to use.Algorithm[2] explains the Left-to-Right scan method.

2.5 Related Work

The concept of Elliptic Curve in Cryptography was introduced by Miller and Koblitz in [7] [8]. In [6], Lenstra and Verheul has shown that 1937-bit key size RSA may provide a similar security as 190-bit key size Elliptic Curve Cryptosystem. In [9], Marc Joye and Sung-Ming Yen has discussed about the left-to-right scanning of bit-by-bit of the key for performing the point multiplication operation.

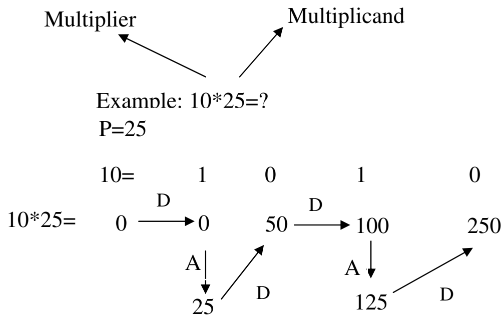


Figure 2.5: Left-to-Right Scan Method

In [10], Sangook Moon has discussed about the scalar point multiplication using the Booth's radix-4 algorithm.

In [11], He, Yajuan and Chang, Chip-Hong used a new redundant binary booth encoding for fast 2n-bit multiplier design. In [12], Villaln Turrubiates did the comparison among the Booth's and Pekmestzi's Algorithm for the multiplication of two numbers. In [13], David J. Malan, Matt Welsh, Michael D. Smith had presented the first known implementation of Elliptic Curve Cryptography Over finite fields for Sensor Networks. In [5] [14], it is analyzed and tested the feasibility of Public-Key Cryptography in Wireless Sensor Networks.

In [15] [16], Rajendra S. Katti and Xiaoyu Ruan has discussed about the partition the integer key into two , reducing the Joint Sparse Form for fast scalar point multiplication. In [17],Hai Yan and Zhijie Jerry Shi has published an article based on the software implementations of Elliptic Curve Cryptosystems. In [4], Pritam Gajkumar Shah, Xu Huang and Charmendra Sharma have discussed about the algorithm based on One's Complement Subtraction for Fast Scalar Multiplicationin ECC for WSN.

2.5.1 One's Complement Subtraction Method

This is a signed representation method, where binary form of the integer k is recoded to its one's complement subtraction form [4]. It is found by the equation

$$(k_{l-1}k_{l-2}...k_0)_2 = 2^l - (\overline{k_{l-1}}\ \ \overline{k_{l-2}}...\overline{k_0}) - 1 \tag{2.3}$$

$$(k_{l-1}k_{l-2}...k_0)_2 = (1(-\overline{k_{l-1}})\ \ (-\overline{k_{l-2}})...(-\overline{k_0} - 1))_2 \tag{2.4}$$

To keep it simple, it is written as

$$C = (2^l - 1) - k; \tag{2.5}$$

where k = Binary Number

l = Number of bits in k

C = One's Complement of the number

However it is not a unique pattern representation as there exists two values for zero (all zeros or all ones), but this is violated as key should be a large prime number and not zero. So, any positive integer is represented by using minimal non-zero bits in its one's complement notation.
The equation(2.5) can be modified as below

$$k = (2^l - 1) - C; \tag{2.6}$$

For example take k=377
$k = (101111001)_2$
$C = (2^l - 1) - k$
$C = (1000000000)_2 - 1 - (101111001)_2$
$C = (010000110)_2$
Therefor, k can be written as $k = (2^l - 1) - C$
$k = 1000000000 - 1 - 010000110$

If every word is splitted out such that it consists only a single one
$k = 1000000000 - 1 - 010000000 - 000000100 - 000000010$
$k = 1\overline{1}0000\overline{111}$

We can observe that hamming weight of recoded integer in OCS approach is lesser than that of original integer. The key length of recoded integer in OCS approach is a bit more than the key length of the original integer. For the above example, the hamming weight of the integer key in OCS approach is 5, whereas in original integer it is 6.

Total computations needed when $k = 377$ in the operation $Q = k.P$ in this approach are calculated as:

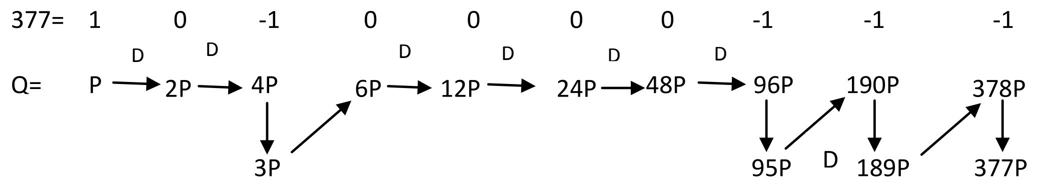


Figure 2.6: OCS Method's Cost Computation

From Figure[2.6], It is observed that total point additions involved are 4 and point doubles are 9.Thus, It results in 48 Scalar Multiplications, 69 Scalar Additions and 13 Scalar Inversions.

Similarly, the total computational cost when no recoding is done on the integer key ,is calculated as:

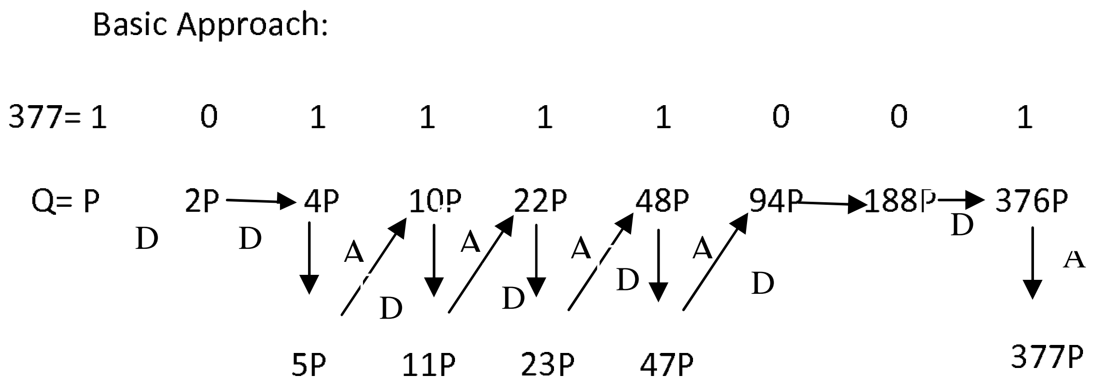


Figure 2.7: Basic Method's Cost Computation

From Figure[2.7], It is observed that the total point additions involved are 5 and point doubles are 8, resulting 47 scalar multiplications, 70 scalar additions and 13 scalar inversions when no recoding of the integer is performed.

Likewise, for odd binary numbers, the number is converted to the required signed-digit representation form keeping all the ones negative expect MSB one. Here in OCS approach, no memory overhead is required because for every recoded integer, each and every bit in the recoded bit pattern is negative other the MSB bit.This approach performs better when there are more than 65% of ones in the integer k.

2.6 Summary

In this chapter, we have seen all the backgroud details of Elliptic Curve Cryptography ,background of point multipliction of ECC, basics of multiplication and some of the various existing methods of point multiplication and related work in brief.

Chapter 3

Proposed Method and Simulation Results

Chapter 3

Proposed Method And Simulation Results

3.1 Proposed Algorithm Based on Booth's Encoding Pattern for Recoding of integer k

A signed binary representation $(k_{l-1}, k_{l-2}, ..., k_0)$ of an integer k is said to be in Non-Adjacent Form(NAF) provided that no two consecutive bits are non-zero.It has certain special proporties like:

- NAF of a number is a unique signed-digit representation.
- NAF defines that no two adjacent non-zeros should be in the bit pattern.
- Hamming Weight of the NAF form is minimal.
- NAF representation contains more zeros than the traditional binary representation of a positive integer
- For regular binary representations of values, half of all bits will be non-zero on average, but with NAF this drops to only one-third of all digits.
- Since every non-zero has to be adjacent to two 0's, NAF representation can be implemented such that it only takes a maximum of l+1 bits for a value that would normally be represented in binary with l bits.

There are various algorithms to obtain the NAF form of an integer, we are confined to Booth's Encoding Pattern to recode the integer k with minimal number of ones in order to reducing the hamming weight.

In hardware, there is no way of storing -1 hence some software approach has

Algorithm 3 Computation of NAF

```
input: $k = (k_{l-1}\, k_{l-2}\, \ldots\, k_0)$ $where$ $k_i \epsilon \{0, 1\}$
ouput: $Z = (Z_{l+1}\, Z_l\, ...\, Z_0)$ $where$ $Z_i \epsilon \{0, 1, -1\}$,
    $i := 0$
while $k > 0$ do
   if $k$ is odd then
      $Z_i := 2 - (k \text{ mod } 4)$; $//Z_i \epsilon \{1, -1\}$ as k is odd
   else
      $Z_i := 0$;
      $k := (k - Z_i)/2$;
      $i := i + 1$;
   end if
end while
```

to be followed to achieve it virtually. Algorithm 4 is for converting the integer k into its NAF form where it stores all the non-zeros as 1 but to differentiate between the 1 and -1, an overhead of memory i.e. equal to the integer key size l and (+1) bits is taken and is used to remember the -1 bits in the NAF form.For a 180 bit key in ecc, another 180 bit overhead is required in our proposed method. It significantly improves the computation needed in this approach than the other existing methods as the number of ones are significantly reduced.

The integer k in Q = k.P is recoded based on the following algorithm

Algorithm 4 Computation of NAF with auxilary memory

```
input: $k \geq 0$
ouput: $Z = (Z_l\, Z_{l-1}\, \ldots\, Z_0)\ \ where\ \ Z_i \epsilon \{0, 1, -1\}$
initialize : $aux = (aux_l\, aux_{l-1}\, \ldots\, aux_0) = 0$,
           $count = 0$;
for $j = l - 1 \rightarrow 0$ do
  if $k_i = 1$ then
    if $count > 0$ then
      if $count = 1$ then
        $aux_{j+1} = 1$;
      end if
      $Z_j = 0$;
      increment $count$ by 1;
    else
      increment $count$ by 1;
    end if
  else if  $count \geq 1$ then
    $Z_j = 1$;
    $count = 1$;
  else
    $count = 0$;
  end if
end for
for $j = i - 1 \rightarrow 0$ do
  $Z_{j+1} = Z_j$;
  $aux_{j+1} = aux_j$;
end for
if $count \geq 1$ then
  $y_0 = 1$;
  $aux_0 = 0$;
else
  $y_0 = 0$;
  $aux_0 = 0$;
end if
```

For example,take k = 377

$k = (101111001)_2$

By applying the Algorithm 4, k is recoded to z

$z = (1010001001)_2$

$aux = (0000101001)_2$

We can observe that hamming weight of recoded integer in our approach is lesser than that of one's complement approach. The key length of recoded integers is same in both the approaches. For the above example, the hamming weight in One's complement approach is 5, whereas in our approach is only 4.

Total computations needed are calculated as

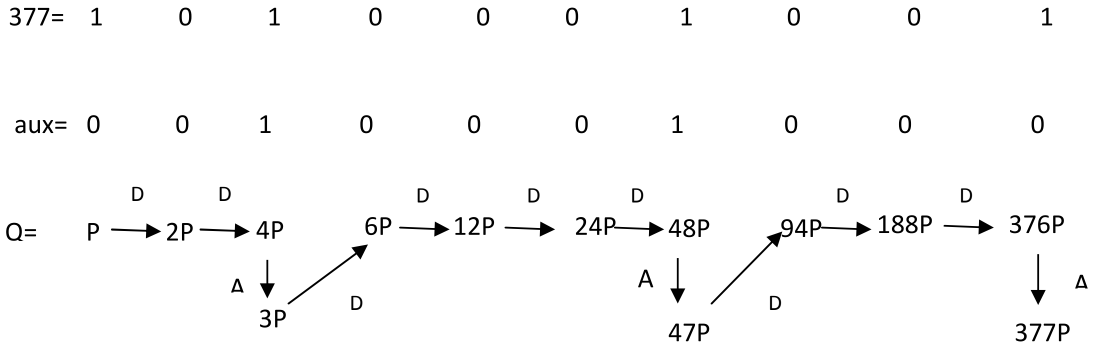


Figure 3.1: Proposed Method's Cost Computation

[Fig 3.1] shows the processing of integer key k by left-to-right scan method and total point additions are 3 and point doubles are 9. Thus, It results in 45 Scalar Multiplications, 63 Scalar Additions and 12 Scalar Inversions which are lesser when compared with OCS method and the basic method.

3.2 Simulation and Results

For simulation, We have considered the elliptic curve cubic equation with the coefficients a, b and p as $2, 3$ and 67 respectively. We have considered a 8-bit key in our simulation paradigm. The key chosen is random and may vary highly with the total number of 1's . Based on the Hamming Weight, The average number of computations (here scalar multiplications,additions and inversions) are plotted against the total number of ones in the original key bit pattern. The three approaches are taken for considered and it has been proven that the proposed approach needed lesser number of computations relatively.

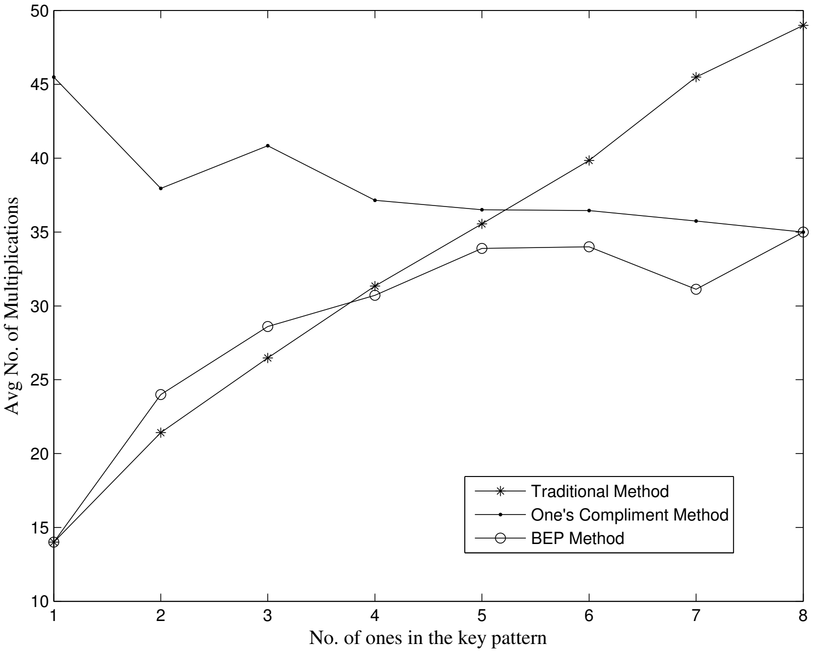


Figure 3.2: Avg No. of Multiplications vs No. of 1's in k

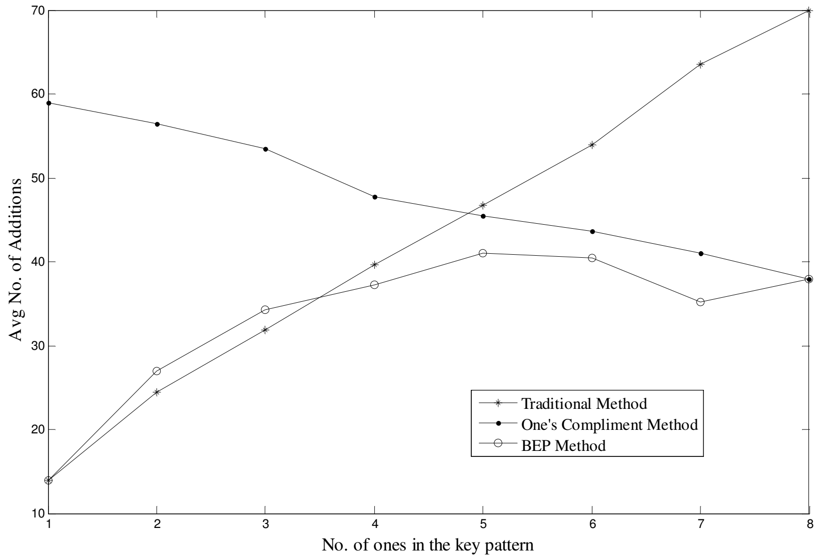


Figure 3.3: Avg No. of Additions vs No. of 1's in k

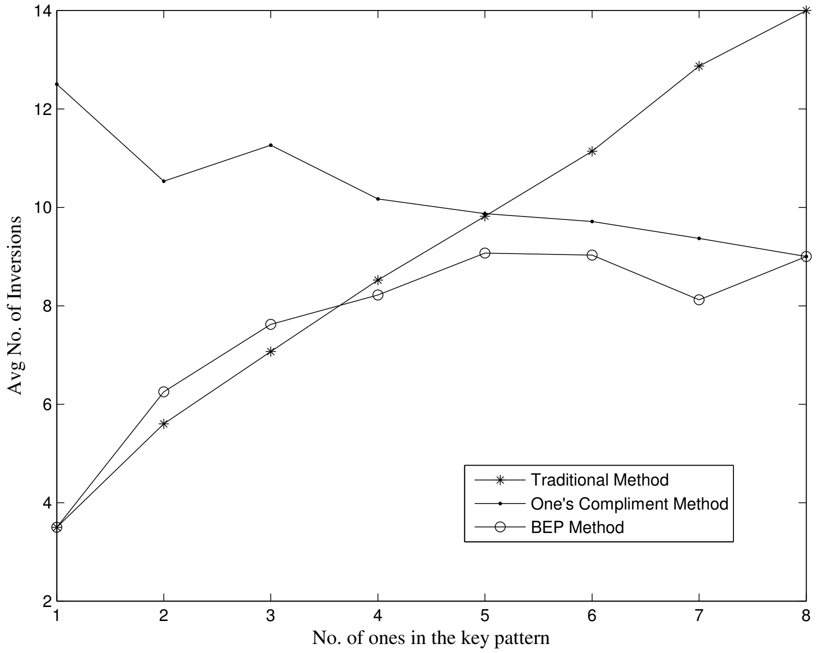


Figure 3.4: Avg No. of Inversions vs No. of 1's in k

3.3 Summary

From Fig[3.2], Fig[3.3] and Fig[3.4], it is observed that the One's compliment approach outperforms the traditional existing basic method when there are more than 65% ones in the key bit pattern. But the proposed approach outperforms both the schemes when the total number of ones in the bit pattern are more than 46% of the bits. The proposed approach drawbacks where there are less number of ones in the key because of the overhead of the extra one-bit added to the recoded integer.

Chapter 4

Conclusion

Chapter 4
Conclusion

In this thesis, We have proposed an algorithm based on Booth's Encoded Pattern to obtain Non-adjacent form with minimal hamming weight by recoding the integer key in Signed-Digit Representation to speed up the point multiplication operation. Minimizing hamming weight reduces the total number of partial scalar operations like additions, multiplications, thus reducing the total computational cost involved in the heavy mathematical Point Multiplication operation of Elliptic Curve Cryptography for wireless sensor network platforms. The 180-bit extra auxiliary memory needed in our approach, considered as overhead to remember the -1's in the recoded pattern for a 180-bit key in Elliptic Curve Cryptosystems does not seems to be a real overhead in Wireless Sensor Network mica motes as it is negligible in 512KB flash memory. This approach provides a very simple way to recode the key reducing the hamming weight, further reducing the computation cost efficiently. The existing OCS method perform better than that of Basic method when there are 65% of 1's in the key bit pattern But our method outperforms the Basic and OCS methods when there are atleast 46% of 1's in the key.

Bibliography

[1] Kossi Edoh. Elliptic curve cryptography on pocketpcs*. *International Journal of Security and Its Applications*, 3(3), 2009.

[2] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *CHES*, pages 119–132, 2004.

[3] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[4] Pritam Gajkumar Shah, Xu Huang, and Dharmendra Sharma. Algorithm based on one's complement for fast scalar multiplication in ecc for wireless sensor network. volume 0, pages 571–576, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[5] Arvinderpal S. W, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks.

[6] Selecting cryptographic key sizes, 2000.

[7] The eliptic curve cryprosystem for smart cards, may 1988.

[8] The elliptic curve cryptosystem, Sept 1997.

[9] Marc Joye and Sung-Ming Yen. Optimal left-to-right binary signed-digit recoding. *IEEE Trans. Comput.*, 49:740–748, July 2000.

[10] Sangook Moon. Elliptic curve scalar point multiplication algorithm using radix-4 booths algorithm, 2005.

[11] Yajuan He and Chip-Hong Chang. A new redundant binary booth encoding for fast 2n-bit multiplier design. *Trans. Cir. Sys. Part I*, 56:1192–1201, June 2009.

[12] Villaln Turrubiates. Comparison among booths and pekmestzi's algorithm for the multiplication of two numbers, June 2003.

[13] David J. Malan, Matt Welsh, and Michael D. Smith. Implementing public-key infrastructure for sensor networks. *TOSN*, 4(4), 2008.

[14] Al-Sakib Khan Pathan and Choong Seon Hong. Feasibility of pkc in resource-constrained wireless sensor networks. *the 11th IEEE Conference on Computer and Information Technology (ICCIT'08)*, 2008.

[15] Rajendra S. Katti and Xiaoyu Ruan. Left-to-right binary signed-digit recoding for elliptic curve cryptography. In *ISCAS (2)*, pages 365–368, 2004.

[16] Xiaoyu Ruan and Rajendra S. Katti. Left-to-right optimal signed-binary representation of a pair of integers. *IEEE Trans. Computers*, 54(2):124–131, 2005.

[17] Hai Yan and Zhijie Jerry Shi. Studying software implementations of elliptic curve cryptography. *Information Technology: New Generations, Third International Conference on*, 0:78–83, 2006.

[18] Chungen Xu and Yanhong Ge. The public key encryption to improve the security on wireless sensor networks. In *Proceedings of the 2009 Second International Conference on Information and Computing Science - Volume 01*, pages 11–14, Washington, DC, USA, 2009. IEEE Computer Society.

[19] Marc Joye. Fast point multiplication on elliptic curves without precomputation. In Joachim von zur Gathen, Jos Imaa, and etin Ko, editors, *Arithmetic*

of Finite Fields, volume 5130 of *Lecture Notes in Computer Science*, pages 36–46. Springer Berlin / Heidelberg, 2008.

[20] Gadiel Seroussi Ian Blake and Nigel Smart, editors. *Elliptic Curves in Cryptography.* Cambridge University Press, Cambridge University Press, The Edinburgh Building, Cambridge CB @RU, UK, 2004.

Dissemination of Work

Communicated

1. Ravi Teja Reddy Levaka,P.M.Khilar, "Algorithm based on Booth's Encoding Pattern for Fast Scalar Multiplication in ECC for Wireless Sensor Networks", in *International conference on frontiers of computer science -2011 (ICFoCS-2011)*, 7-9th August, 2011, Bangalore, India.(**Communicated**)