Optimizing point Doubling Operations in ECC Z_p

A.Sakthivel
ASSISTANT PROFESSOR
Adithya institute of technology
India

R.Nedunchezhian
PROFESSOR
Sri Ramakrishna Engineering
College, India

ABSTRACT

Today a wireless network has minimum power consume and less security access. Based on these the suitable way of providing a security system for wireless application is to select the Elliptic Curve Cryptography. But this public key cryptography has required more number of clock cycles to compute its point operations. One of the point operations called point multiplication requires a lot of clock cycles to compute result. This proposed technique reduces the number of clock cycles of point multiplication for parallel processing by reducing number of dependent and independent operations.

General Terms

Wireless network,security system and dependent and independent operations.

Keywords

Public key cryptography, Elliptic curve cryptography, Point multiplication, clock cycles, parallel processing.

1. INTRODUCTION

Some of the biggest challenges in the field of wireless communication are to provide a high security for various application services, to reduce power consummation of various processes and low bandwidth usage for communication media[2]. So it is very important to find out suitable asymmetric key cryptosystem to offer the same. In this case the Elliptic curve cryptosystem is more suitable than any other cryptosystem such as RSA,Elgamal,NIST and Rabinson[7]. Because the ECC has a set of merits called as high security, less key size, low power consummation and low bandwidth requirements to support wireless communications when it compares with other public key cryptosystem[12].

This Elliptic Curve Cryptosystem has two main categories called as ECC over prime field (Zp) and ECC over binary field (2p). In these types, ECC over prime field is used for software implementations and ECC over binary field for hardware implementations. This paper suggests a new approach for ECC over prime field implementation and analyzes the same. This Cryptosystem consists of Elliptic Curve (EC) Cryptography and Cryptanalysis. The EC Cryptography defines the encryption which converts the original text into the secret text and the decryption the secret text into original text. In addition to, the EC Cryptanalysis analyzes the strength of ECC based on various constraints and parameters.

The above mentioned Elliptic curve cryptography and Elliptic curve cryptanalysis are implemented by using a set of point operations such as point addition, point subtraction, point multiplication, point division, point inversion and point doubling. In these operations, the time complexity of point doubling is higher than any other point operations on Elliptic Curve. And also the point doubling is the heart part of the ECC. So it is necessary to find out optimized implementations for point doubling. Some of the already available implementations do not support code scheduling for parallel processing on arithmetic operations. Hence it is important for future to find out best implementations for point doubling operations which improves the performance of the ECC.

For the above mentioned reason, the proposed paper is organized as follows. Section 2 explains the basic background of ECC and the mathematical foundation of ECC. Followed by, section 3 describes the proposed work in detail. The section 4 analyzes the performance of ECC briefly and finally the conclusion of this work is submitted in the section 5.

2. BACKGROUND

2.1 ECC over Prime Field

An elliptic curve over a finite field K is defined by the general Weierstrass equations:

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_5 \qquad (1)$$

where a_1 ,a_3 ,a_2 ,a_4 ,a_5 Є E

A set of pairs(x,y) which solves (1) and the point at infinity denoted by 'O' called as additive identity for the abelian group which are described in appendix are used to implement elliptic curve cryptography[5]. The elliptic curve can be analyzed over polynomial basis(10),dual basis(2), triangular basis(6) and redundant basis(3). The finite field over polynomial basis which is described in appendix is mostly suitable for cryptographic software application[3][4]. Besides the general equation(1) should satisfy the following constraints:

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

x,y,a and b Є $GF(2^n)$ and $\Delta=4a^3+27b^2\neq0$

It is called as non-super singular form equation and points are called as affine and projective co-ordinates[3][5][7]. Co-ordinates are manipulated by using a set of operations called as point addition and point doubling[17]. The point addition rules over Er(a,b) is the basic operation on Elliptic curve. P,Q

Є Er(a,b) where P and Q points on Elliptic curve straight line and the result of P+Q is also on same line.

if P=(x1,y1), Q=(0,0) Є E(a,b) then R=P+Q=P+O=P
where O is origin points (3)
if P=(x1,y1),Q=(x1,x1 + y1)Є E(a,b) then
R=P+Q=P+(-P)=O
where -P is inverse of P (4)
if P=(x1,y1), Q=(x2,y2) Є E and P=Q then
R=P+Q=(x3,y3)
where x3=λ-x1-x2, y3=λ(x1-x3)-y1
λ= (y2-y1)/(x2-x1) if(P≠Q) (5)
λ=(3x12+a)/2y1 if(P=Q) (6)

The point addition rules over Er(a,b) is the basic operation on Elliptic curve. The importance of ECC is to compute the point multiplication Q=kP, where k is a scalar value and P is a point on the elliptic curve. The point multiplication is defined by using two operations of ECC over prime field called as point addition and point doubling. The number of points in Er(a,b) is approximately equal to the number of elements in Zr, namely P elements and bounded by $r+1-2\sqrt{r} \leq N \leq r+1+r\sqrt{p}$ which is also known as scalar multiplication[1][4][9]. It is denoted in equation (7) as shown figure 1.

$$kP= P+P+P+P...(k-1)P+kP \qquad (7)$$

```
procedure linearmultiplication (Point P,Integer k)
1.Integer I, Q0=(0,0)
2.I=1
3.if(I⩽k)
4.compare P with Q0.
5.compute and update P and Q0 by using equation (3),(4),
(5)or(6)
6.I=I+1 goto step 3.
7. return P which has output kP
```

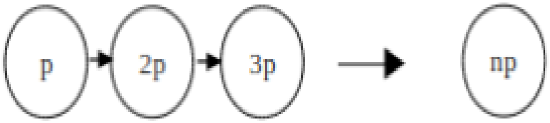

Fig 1: Linear Multiplication

2.2 Data Structure for Software Scheduling

From the literature survey shows that there is no optimal scalar multiplication implementation for parallel processing. In the paper, one of the data structure binary trees and divide and conquer are used for scalar multiplication[12]. A divide and conquer algorithm works by iteratively breaking down multiplication into two or more sub-problems of the same type, until problems become simple to compute directly. After the solution of problems are combined to find out the required computation. According to the divide and conquer strategy, the binary tree and skew binary tree structures are used to find out the result of computation[10]. There are four types of dependences existing in Linear scalar multiplication which is shown in figure 2 namely Data dependences, Name Dependences, Control Dependences and Loop carried Dependences which affect the speed of computation. The name Dependence means that two or more points refer the same name and a point is dependent on another point called as Data dependences. A control dependence determines the way of computing points based on four constraints which are mentioned in equation 3,4,5 and 6. Finally the analysis of loop-level parallelism focuses on determining whether point accesses in later iterations are dependent on point produced in earlier iterations or not, called as a loop carried dependence.

3. PROPOSED TECHNIQUE

The scalar value of equation (7) is used to create binary tree and skew tree nodes and each node has a point value. Finally the summing of skew tree node value and binary tree node value computes kP value shown in figure 2. The k value is assumed as 15. This type of computation is called as divide and conquer strategy[14]. When the k value is divided by 2 in every time, quotient and remainder values are obtained. The binary tree is created by using quotient value and points are used to compute point doubling operation based on the formula (6). And subsequently the skew tree is also formed by using reminder value and points to compute point addition operation based on the formula (7). Finally these two trees are summed by using formula (4),(5),(6) or (7) to compute point doubling for kP. Procedures are explained as follows:

```
Procedure PointCompute(Point P,Integer k)
Point P1=(0,0),P2=(0,0);
1.If k=1 then
    1.1 One time of P
2.If k>1 then
  2.1 Q←k/2
  2.2 if (Q>0) then
    2.2.1 P1=call PointDoublingBinary(Point P)
    2.2.2 P=P1
  2.3 R←k mod 2
  2.4 if(R=1) then
    2.4.1 P2=call PointDoublingSkew(Point P)
    2.4.2 P=P2
3. k=k/2 and goto step 2.
4. call Procedure for PointSummazation(Point P1,Point P2)

Procedure PointDoublingBinary(Point P, Quotient Q)
Point Sum=(0,0).
1. Find out SUM=P+SUM based on equation (3),(4),(5) or (6)
   and return SUM value.

Procedure PointDoublingSkew(Point P,Remainder R)
Point Sum=(0,0).
1. Find out SUM=P+SUM based on equation (3),(4),(5) or (6)
   and return SUM value.

Procedure for PointSummazation(Point P1,Point P2)
Point Sum=(0,0).
1.Find out SUM=P1+P2 based on equation (3),(4),(5)
   and (6).
 2.Display point doubling value for kP.
```

The above proposed methodology minimizes number of loop carried dependence, data dependence, control dependence and register dependence which helps to improve code scheduling on software to minimize the different hazards and stalls during execution. But the linear scalar multiplication Equation 7, P is repeated k times alternatively to perform the different case of point doubling operation[5]. During the processing, the recent iteration is always dependent on early iteration known as loop carried dependence[11]. This type of Data dependence creates N-1 times of Hazards and stalls(WAR,RAW,WAR) to affect loop level parallelism[8][18]. If it is not optimized, it will affect the performance of the computing. But the proposed divide and conquer strategy using trees will reduce number of dependence into $\log_2 N+1$ or $2\log_2 N+1$ dependent operations[9].

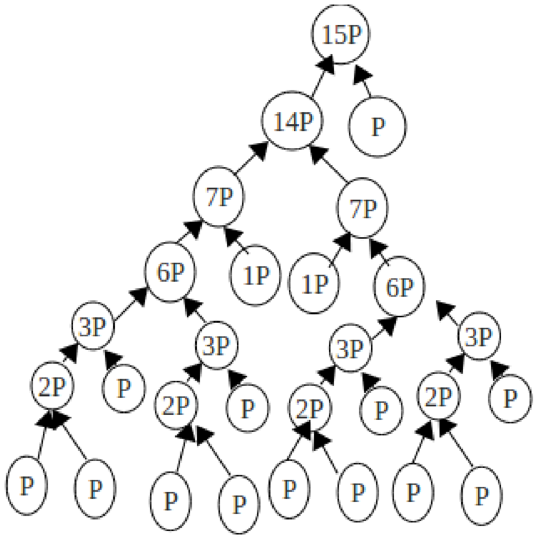

Fig 2. The Point multiplication of kP based on conquer process.

4. PERFORMANCE ANALYSIS

An equation $y^2=x^3+ax+b$ is considered to support a set of points on Elliptic Curve. Then $E_p(a,b)$ is defined by using $E_{11111}(1,1)$. The point (x=0,y=1) is taken from the set to compute point multiplication for both linear scalar and the proposed binary tree multiplication[21][22]. In proposed case, there are three cases to analyze point doubling by using k value.

First Case is called as best case. Because there is no remainder for k value in all iterations. It means that $k= 2^N$ where N>0. Because it takes only $\log_2 N$ times to compute kP based on Quotient point computation and no need to compute skew tree computation. The second case is called as worst case. Because there is a remainder and quotient values for k in all iteration. So the k value is in the form of 2^N-1 where N>0. The k value takes only $\log_2 N$ times to compute kP based on quotient point computation as well as remainder point computation. The time complexity of this case is defined by $2\log_2 N$ times.The third case is called as average case. Because there is a remainder for k values in some iterations and no remainder for some other iterations. The k value is in 2^N or 2^N-1. The computation time of this case is defined by $\log_2 N$+Prob{$\log_2 N$}times. Three cases of time complexities are redefined by $\log_2 N+1$,$2\log_2 N+1$ and $\log_2 N$+Prob{$\log_2 N$}+1. The value 1 denotes the final computation of combining quotient point computation and remainder point computation to compute kP.

The best and worst cases of proposed methodologies are compared with linear scalar point doubling. Both are implemented language and computation values are measured in terms of clock pulses as shown in the table 1.

Table 1. Comparison between Linear multiplications and Tree Point multiplications

calculates kP		Clock pulses		
i	2^i	Best	Worst	Linear
1	2	0	0	0
2	4	1	1	3
3	8	2	3	4
4	16	3	5	6
5	32	4	6	13
6	64	5	7	24
7	128	6	9	46
8	256	6	10	91
9	512	7	11	191
10	1024	7	11	380
11	2048	8	12	748
12	4096	9	13	1503
13	8192	9	14	3054

The Comparison among linear method, the proposed worst case and best case is shown in Figure 3. This concludes that the proposed methods should decreases the number of clock pulses to reduce computation time and increases the performance of the system which is proved by Amdahl's law attached in appendix. In the scalar multiplication the point doubling is computed by using an Equation 5. frequently and an Equation 6 rarely. But in proposed method is computed in vice versa. Normally the Equation 6 computation time is lesser than the Equation 5 because of number of operations. So the proposed methodology is better than the scalar multiplication.

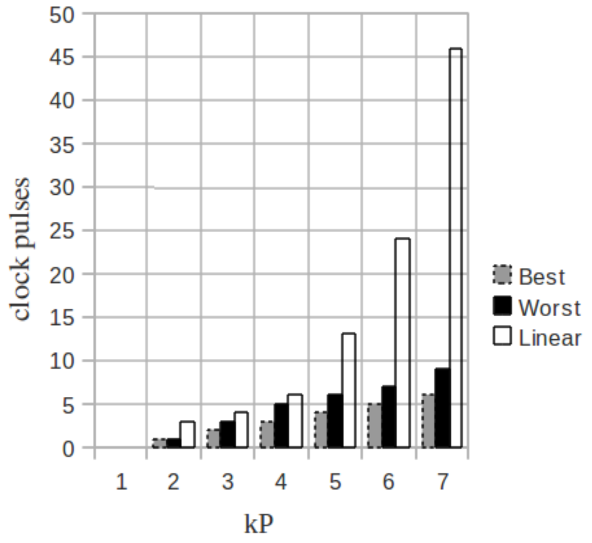

Fig 3: Comparison between Linear multiplications vs Tree Point multiplication

Then the proposed method is analyzed in different perspective with graph in shown figure 3. In this graphs, the x-axis denotes k values in the form of 2i and y axis numbers of clock pulses needed to compute kP[12].

5. CONCLUSIONS

The traditional ECC uses Linear scalar point multiplication to perform encryption or decryption[19]. The consequences of using this approach needs more clock pulses. The proposed work reduces number of clock pulses, power consumption[13] and increases the performance of ECC. This software scheduling converts dependent into independent iteration and reduces number of computation and eliminates data hazards and stalls for arithmetic operations[15]. This computation is very much useful when it is trying to enhance secure e-Mail applications such as PGP or S/MIME , SMS, security protocol design, e-Business, e-Banking and etc[16][22]. The proposed methodology is to utilizes hardware units minimum number of times to find out point doubling and the life lime of hardware units.

APPENDIX

Amdahl's law

Amdahl's Law states that the performance improvements to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used. It is used to defines Speedup of the machine.

The CPU Efficiency

CPU time=CPU clock cycles for a program×clock cycle time

Abelian group

A Finite Field is defined with ring and multiplicative structure. The ring structure is formed by using the characteristics of abelain group. where G → a group with set of elements, {a,b,c,I,a-1,b-1,c-1} Є G. and * → operator over G[9].

ACKNOWLEDGMENTS

Authors would like to thank the esteemed reviewers for their valuable comments which have made solving point doubling using the Linear Scalar multiplication on elliptic curve equations and this work is a replacement for the simplified proposed methodology.

REFERENCES

[1] Jyu-Yuan Lai & Chih-Tsun Hung. 2008. Elixir: High-Throughput Cost-Effective Dual-Field Processors and Design Frame work for Elliptic Curve Cryptography. IEEE Transactions on very Large scale Integration Systems,*vol. 16*,No.11.

[2] Sandro Bartolini & Roberto Giorgi. 2008. Effects of Instruction-Set Extension on an Embedded processor: A case study on Elliptic-Curve Cryptography over GF(2^m). IEEE Transaction on computers, vol.57,No.5 .,pp.289-302.

[3] William, N., chelton, mohammed Benaissa. 2008. Fast Elliptic-Curve Cryptography on FPGA, IEEE Transaction on very Large Scale Integration Systems. vol.16,No.2, pp.198-205.

[4] Patrick Longa and Ali Miri. 2008. Fast and Flexible Elliptic Curve Cryptography point arithmetic over Prime fields. IEEE Transactions on computers, vol.57,No.3,pp.289-302.

[5] Kimmo Jarvinen &Jorma Skytta.2008.On Parallel of High Speed Processors for Elliptic Curve Cryptography.IEEE Transcations on very Large scale Integration Systems,vol 16,No. 9.

[6] Kazuo Sakiyama, Lejila Batina, Bart Preneel AND Ingrid Verauwhede, 2007. Multicore Curve Based Cryptoprocessor with Reconfirurable Modular Arithmetic Logic Units over GF(2n). IEEE Transactions on computers,vol.57,No.9. ,pp1269-282.

[7] Sining Liu,Brian King & Wei Wang. 2007. Hardware Organization to achieve High Speed Elliptic Curve Cryptography for Mobile Devices. Springer Science+-Mobile New Appl(2007), vol 12,pp271-279.

[8] John, L., Hennessy AND David, A., Patterson.2006. Computer Architecture a Quantitative Approach. Elsevier,4th Edition.

[9] William Stallings, 2003. Cryptography and Network Security. PHI, 4th Edition.

[10] Mark Allen Weiss. 2006. Data Structures and Algorithm Analysis in C, Pearson Education, 2nd Edition .

[11] Pradeep Kumar Mishra. 2008. Pipelined Computation of Scalar Multiplication in Elliptic Curve Cryptosystems. IEEE Transactions on computers, vol 55 ,No.8,pp1000-1010.

[12] David, J.,Malan, Matt Welsh, & Michel, D., Smith. 2004. Implementing Public Key Infrastructure for Sensor Networks. ACM Transactions on Sensor Networks,vol.4 ,No. 4. Article 22.

[13] Catherine H.Gebotys, 2004. Design of Secure Cryptography Aginst threat of Power-Attacks in DSP-Embedded Processors,ACM Transactions on Embedded Computing Systems, Vol 3,No. 1,pp.92-113.

[14] Rodrigo Roman, Cristina Alcaraz and Javier Lopez. 2007. A survey of Cryptographic Primitives and Implementations for Hardware-Constrained Sensor Networks Node. Springer Science+-Mobile New Appl(2007), vol 12,pp231-244.

[15] Volker muller, 1998. Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two*, J. Cryptology 11: 219–234.

[16] N. P. Smart, 1999. Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic,J.Cryptology,12: 141–151.

[17] Daniel V. Bailey and Christof Paar, 1999 Efficient Arithmetic in Finite Field Extensions with Application in Elliptic Curve Cryptography, J. Cryptology 12: 193–196.

[18] Steven D. Galbraith, 2002. Elliptic Curve Paillier Schemes, J. Cryptology, 15:129–138.

[19] Elisavet Konstantinou, Aristides Kontogeorgis, Yannis C Stamatiou and Christos Zaroliagis. 2002. On the Efficient Generation of Prime-Order Elliptic Curves, J. Cryptology,15: 129–138.

[20] M. Barbosa, A. Moss, and D. Page 2002. Constructive and Destructive Use of Compilers in Elliptic Curve Cryptography, J. Cryptology ,5:129-138.

[21] V.Gayoso Martinez, F.Hernandez Alvarez, L.Hernandez Encinas and C. Sanchez Avila 2011. Analysis of ECIES and Other Crypto systems Based on Elliptic Curves.

[22] V.Gayoso Martinez, F.Hernandez Alvarez, L.Hernandez Encinas and C. Sanchez Avila. 2010. A Comparision of the standardized Versions of ECIES.A, IEEE International Conference on IAS. Proceeding. pp.1-4.