ISSN No. 0976-5697

Volume 2, No. 5, Sept-Oct 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Analysis of Finite Field in Elliptic Curve Cryptography for Software Implementation

Ms. Amita Rathee*
Master of Technology Computer Engineering
Prabhu Dayal Memorial College of Engineering
Bahadurgarh, Haryana,India.
amitarathee12@gmail.com

Prof. (Dr.) Paramjit Singh
Director Academics
Prabhu Dayal Memorial College of Engineering for Women
Bahadurgarh, Haryana,India.
drparamjit@pdmce.ac.in

Abstract: Finite field is the field with finite number of elements. In Elliptic Curve Cryptography finite field can be prime field or extensive field. In this paper we will study the finite fields those are suggested and analyze that how general prime field are used and are much more compatible with software implementation then other existing. Existing study suggests to use binary fields as they have low cost of implementation but we are suggesting to use general prime field though their implementation cost is little high then those of binary but they desire less computational power as compared to them. In this paper we are suggesting the prime field that would be future of finite field in elliptic curve cryptography.

Keywords: Elliptic Curve Cryptography, Extensive Fields, Finite Field, Prime Field.

I. INTRODUCTION

Finite Fields are the fields those have finite number of elements. Finite fields are categorized as Prime fields and Extensive fields. Elliptic Curve Cryptography (ECC) was first proposed by Victor Shoup Miller and Neal Koblitz independently in 1985. The strength of Cryptographic systems lies on how hard it is to decrypt a message. Finite fields are the basis for selection of points on the elliptic curve selected. Finite field plays an important role in efficient architecture design and implementation of ECC In this paper we will discuss the basic concepts of finite fields.

ECC consumes less memory and hardware resources to implement that why it can be used with smaller devices. The basis of the strength of ECC is the hard problem used in it that is known as Elliptic Curve Discrete Logarithm Problem (ECDLP) which is derived from Discrete Logarithm Problem (DLP). The efficiency of every cryptosystem based on the hardness of the hard problem used. There are many public key cryptosystems some of them depends on Factoring Big Number Problem that operates on integer fields. But Elliptic Curve Cryptosystems are based on Elliptic Curve over Finite Field.

Finite Field is the base for the architecture of ECC as it provides the some of basic domain parameters those are used in various parts of ECC, such as in key generation, scalar multiplication, signature generation, signature verification, encryption and decryption.

The Structure of remaining paper is as follows: Section 2 will discuss what are finite fields? What kind of finite fields are available? Section 3 will discuss what finite field operations are there? Section 4 will discuss what elliptic curves are? Section 5 will suggest which finite field would be most appropriate for software implementation of Elliptic Curve Cryptography. Finally the conclusion followed by references.

II. FINITE FIELD

Finite Field is a field over finite number of elements. Finite field is also known as Galois Field named after founder of finite field theory Evariste Galois, French mathematician.

The points on Elliptic curve form an abelian group. Abelian group is also known as commutative group. It is a group in which the result of applying the group operation to two group elements does not depend on their order. Abelian group is named after Niels Henrik Abel. An Abelian group (G,*) consists of a set G with a binary operation *: G × G —> G satisfying following properties:

A. Associativity
   a * (b * c) = (a * b) * c, for all a,b,c Є G.
B. Existence of Identity
   There exists an element e, e Є G, such that
   a * e = e * a = a, for all a Є G.
C. Existence of Inverse
   For each a Є G, there exist an element b Є G, called inverse of a,
   such that a * b = b * a, for all a,b Є G.
D. Commutativity
   a * b = b * a, for all a,b Є G

Finite fields proposed for use in public key cryptosystems are Prime Fields, and Extensive Fields[1][2]. As we know implementation of ECC depends on its finite field and different fields have different implementation methods.

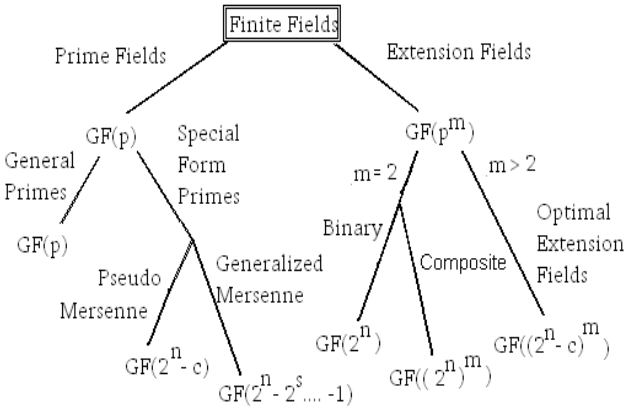

Figure 1: Finite Field Classification

The order of a finite field is the number of elements in the field. There exists a finite field F with order q, if and

only if q is a prime power, i.e. $q = p^m$, where p is prime number called characteristic of field F, and m is positive integer. If m = 1, then F is a Prime field. If m > 2, then F is an Extensive field[3].

III. FINITE FIELD ARITHMETIC

The Finite field in Elliptic curve cryptography supports two operations addition and multiplication. Subtraction and Multiplication operations are defined in terms of addition.

There are some addition rules for points on elliptic curve under finite fields. The addition rules for two points P and Q are:

a) O + P = P, and P + O = P, where O is Identity element.
b) – O = O.
c) If $P \neq O$, then – P = (x_1, - y_1 - a_1 x_1 - a_3).
d) If Q = - P, then P + Q = O.
e) If $P \neq O$, $Q \neq O$, $Q \neq -P$, then let R be third point of intersection of either line intersecting P, Q.
f) If $P \neq Q$ or P = Q, then P + Q = - R.

A. Point Addition

Point Addition is the operation of adding the points of elliptic curve to find another point on the curve.

The Additive property defined geometrically on an elliptic curve is P + Q = R, given the points P and Q on elliptic curve find the point R on the curve. Elliptic Curve groups are additive groups, that is, their basic operation is addition.

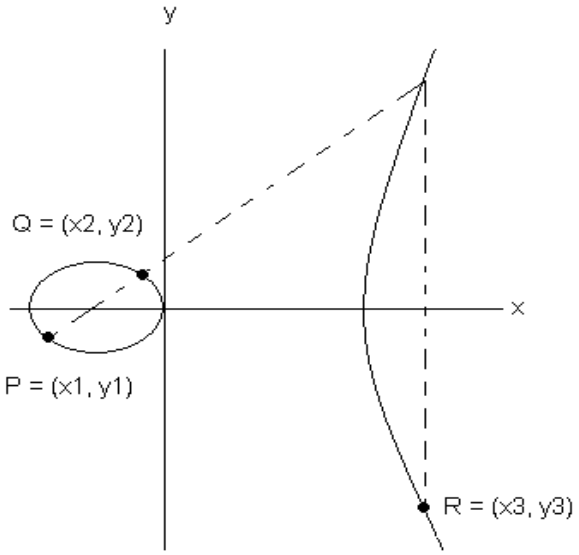

Figure 2: Point Addition

The negative of a point P on elliptic curve is its reflection on x-axis is –P on same curve, and for each point P on an elliptic curve, -P is also on same curve. To add two distinct points P(x1,y1) and Q(x2,y2) on an elliptic curve, such that P is not –Q that means P should not be reflection of point Q on the curve, draw a line through these points. This will cut the elliptic curve in exactly one more point –R. Then point –R is reflected to R to have the resultant point on the same curve.

The law of addition in elliptic curve group for different points P and Q on elliptic curve is P + Q = R such that

$$P+Q=\left[\frac{y_2-y_1}{(x_2-x_1)^2}-x_1-x_2,\ \frac{(y_2-y_1)(x_1-x_3)}{x_2-x_1}-y_1\right]$$

where x3 is the x coordinate of P + P that is the doubling of point P.

To add a point P and its reflection –P on an elliptic curve, draw a vertical line which doesn't intersect curve at any third point on the elliptic curve, thus they can't be added as previously. So, elliptic curve group including infinity point O.

By definition, P + (-P) = O, i.e., P = P + O, where O is additive identity of elliptic curve group.

Doubling a Point P on an elliptic curve: To add a point $P(x_1, y1)$ on an elliptic curve to itself, a tangent line to curve is drawn at point P. If $y_1 \sim= 0$ then tangent line intersects the elliptic curve at exactly one other point –R on the same curve. The –R is reflected to x-axis to R. This is doubling a point P, and the law of doubling a point on an elliptic curve group is defined by: P + P = 2P = R such that

$$P+P=\left[\left(\frac{3x_1^2+a}{2y_1}\right)^2-2x_1,\ \frac{(3x_1^2+a)(x_1-x_3)}{2y_1}-y_1\right]$$

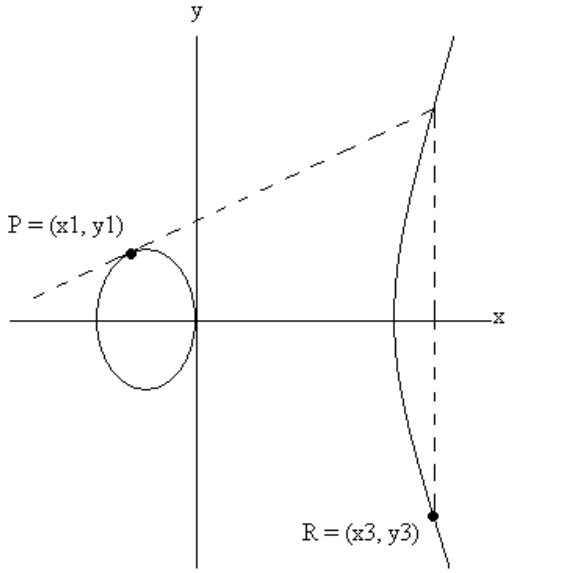

Figure 3: Doubling Point

If y_1 = 0 then tangent from point P will always be vertical and doesn't intersect elliptic curve at any other point.

By definition, 2P = O for such a point P.

B. Point Multiplication

Multiplication in a finite field is the multi-plication modulo, an irreducible reducing polynomial, used to define the finite field. In Point Multiplication, a point P on elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on same elliptic curve.

Q = kP = P + P + …… + P (k times addition)

Point Multiplication is achieved by two basic operations of elliptic curve operations: Point Addition and Point Doubling. Point Addition or Point Doubling can be used repeatedly to find point multiplication but it would be more complex as the number of computations rise.

We used the Modified Weierstrass form of elliptic curve E: $y^2 = x^3 + ax + b$ in our work, where the order of elliptic curve n is 3. So the scalar for point multiplication is chosen in range 0 to 2. The scalar we used in our work is generated randomly for point multiplication. It is used while key generation to find a point Q on the elliptic curve given with point P.

The sample code in MatLab is as follows:

```
a=round(0+(10-0)*rand(1));
b=round(0+(10-0)*rand(1));
c=((4*a*a*a)+(27*b*b));
if c==0
return;
else
disp('Coefficients selected for Elliptic Curve');
disp(a);
disp(b);
x1=round(1+(5-1)*rand(1));
y1=round(sqrt((x1*x1*x1)+(a*x1)+b));
disp('Coordinates of P');
disp(p1);
disp(p2);
k=round(1+(3-1)*rand(1));
disp('Randomly selected integer k = ');
disp(k);
x2=(k*x1);
y2=(d*y1);
disp('Coordinates of Q');
disp(x2);
disp(y2);
end
```

The above code is for key generation from elliptic curve selected the curve will be generated by selecting the coefficients of curve randomly and point multiplication is done to find the point Q on the curve after generating the point P randomly. The scalar used to find point Q is k that ranges from 0 to 2.

We suggest using Point Multiplication directly multiplying with scalar k and not by repeatedly adding point to itself, as it will reduce the code for software implementation and also reduce the complexity and time to compute.

IV. ELLIPTIC CURVES

Elliptic Curves are mathematical structures from number theory and algebraic geometry those are not ellipses but non-singular projective curves based on cubic equation that was given by Karl Weierstrass [3].

Weierstrass form of Elliptic curve is:

$$E : y^2 + axy + by = x^3 + cx^2 + dx + e$$

where $a,b,c,d,e \in F$

F is finite field over which curve is defined.

Discriminate of the curve is defined as

$$\Delta = d_1^2 d_4 - 8 d_2^3 - 27 d_3^2 + 9 d_1 d_2 d_3$$

where

$d_1 = a + 4b$

$d_2 = 2d + ac$

$d_3 = c^2 + 4e$

$d_4 = a^2e + 4be – acd + bc^2 – d^2$

and $\Delta \neq 0$

For finite field F = GF(p)

Modified Weierstrass form is

$$E: y^2 = x^3 + ax + b$$

where $\Delta = -16(4a^3 + 27b^2)$

and $4a^3 + 27b^2 \neq 0$

The discriminate of curve should not be equal to zero so that it can form a group that is abelian group. The elliptic curve has the basic condition for randomly generated curve that is the discriminate should not be zero. If discriminate is zero then try for another curve.

V. SUGGESTED FINITE FIELD FOR ELLIPTIC CURVE CRYPTOGRAPHY

According to reference [5], if finite field used in Elliptic Curve Cryptography is $GF(2^m)$ and is implemented with hardware only will not be very good and if implemented with software that will be good but would be slower in comparison to that of hardware implementation. If finite field used in Elliptic Curve Cryptography is GF(p) that will be suitable with software but not with hardware. If finite field used in Elliptic Curve Cryptography is $GF(p^m)$ that will be suitable for software but will be hardware dependent. So the finite field GF(p) will be the future trend of Elliptic Curve Cryptography.

According to reference [6], $GF(2^m)$ is well suited for hardware implementation while software implementation is optimized with GF(p). The reason is that with the general prime field we need less size for key the in binary finite field or with other available. It supports software implementation and requires less memory for computation.

Table 1: Comparison of Finite Fields in ECC and RSA

Elliptic Curve Cryptography		RSA	Security Level
F_P |P| [bits]	F_{2^m} m [bits]	|n| [bits]	
112	113	512	56
128	131	704	64
160	163	1024	80
192	193	1536	96
224	233	2048	112

The above table clearly shows that there is minor difference between the key sizes of Prime field and binary field. But this fraction of difference is valuable when the system need more security with lesser available key size to save resources and reduce number of computation to carry out to reduce the overhead.

The prime finite field F_P have characteristics that is P >3 with curve $y^2 = x^3 + ax + b$ and $\Delta = -16(4a^3 + 27b^2) \neq 0$, that less complicates it whereas fields F_{2^m} have two characteristics, first is if $a \neq 0$ that is a non -singular curve $y^2 + xy = x^3 + ax^2 + b$ and $\Delta = b \neq 0$, second is if a = 0 that is singular curve $y^2 + cy = x^3 + ax + b$ and $\Delta = c^4 \neq 0$. The simplicity of Prime field makes it more preferable and with this we maintain same level of security as with other with less key size.

VI. CONCLUSION

In this paper we analyzed that Finite Field are the basic architecture of Elliptic Curve Cryptography. We briefly introduce what Finite Fields are? What kind of Finite Fields are available for Elliptic Curve Cryptography? What are Elliptic Curves? The finite fields are important to reduce the computational complexity. In the available finite fields,

general prime fields are most appropriate field for Elliptic Curve Cryptography. We have discussed the aspects of finite fields discussed earlier and also provided the sample code which shows how important is the finite field selection for optimized Elliptic Curve Cryptography for software implementation.

VII. ACKNOWLEDGMENT

I would like to thank my guide Prof. Paramjit Singh for his helpful suggestions throughout this work of piece.

VIII.REFERENCES

[1]. Christof Paar, "Implementation Options for Finite Field Arithmetic for Elliptic Curve Cryptosystems", Invited presentation at 3rd Workshop on Elliptic Curve Cryptography (ECC '99).University of Waterloo, Waterloo, Ontario, Canada, 1999.

[2]. Bimal Kumar Meher, "A Study of Suitability and Effectiveness of Various Implementation Options of Finite Field Arithmetic on Elliptic Curve Cryptosystems", International Journal of Computer Theory and Engineering, Vol. 1, No. 4, Department of Information Technology, Silicon Institute of Technology, Bhubaneswar, Orissa, India, October 2009.

[3]. M. Brown, D. Hankerson, J. Lopez, A. Menezes, "Software Implementation of the NIST Elliptic Curves Over Prime Fields", Dept. of C&O, University of Waterloo, Canada, Nov, 2010.

[4]. Andrew Byrne, Francis Crowe, William P. Marnave, "SPA resistant elliptic curve cryptosystem using addition chains", High Performance Systems Architecture, Vol. 1, No.: 2, 2007.

[5] Robinson J A, Liang V M, Chambers J A, M,et al. Computer user verification using login string keystroke dynamics[J]. IEEE Trans on Systems, Man, and Cybernetics ,Part A: Systems and Humans , 1998, 28(2) : 236 – 241.

[6] Guerric Meurice de Dormale, Philippe Bulens, Jean-Jacques Quisquater, "Collision Search for Elliptic Curve Discrete Logarithm over GF(2*m*) with FPGA", Belgium.