Study of Finite Field over Elliptic Curve: Arithmetic Means

Samta Gajbhiye
Associate Professor, CSVTU
CSE dept
SSGI, Bhilai(C.G)

Sanjeev Karmakar
Associate Professor, CSVTU
MCA Dept
BIT Bhilai [CG]

Monisha Sharma
Professor, CSVTU
ETC Dept
SSGI,Bhilai(C.G)

ABSTRACT
Public key cryptography systems are based on sound mathematical foundations that are designed to make the problem hard for an intruder to break into the system. Number theory and algebraic geometry, namely the theory of elliptic curves defined over finite fields, has found applications in cryptology. The basic reason for this is that elliptic curves over finite fields provide an inexhaustible supply of finite abelian groups which, even when large, are amenable to computation because of their rich structure. The first level is the mathematical background concerning the needed tools from algebraic geometry and arithmetic. This paper introduces the elementary algebraic structures and the basic facts on number theory in finite fields. It includes the minimal amount of mathematical background necessary to understand the applications to cryptology. Elliptic curves are intimately connected with the theory of modular forms, in more than one ways. The paper gives a brief introduction to modular arithmetic, which is the core arithmetic of almost all public key algorithms. . The ultimate goal of the paper is to completely understand the structure of the points on the elliptic curve over any field *F* and being able to find them.

Keywords
Abelian group, cyclic group, Binary Field, Prime Field, Elliptic Curve.

1. INTRODUCTION
Modern algebra, like various other branches of mathematics, offers conceptual models for design, analysis, and proof for wide range of problems. The most constrained structures of modern algebra are fields, and after them are rings. At the simplest end of spectrum is the subgroup structures monoids, semi-groups (subsets of group, eg: not having an inverse, such as operations on strings, or languages, such a concatenation of strings. Strings do not have inverses). Without an inverse a decryption is not possible for an encryption. Hence group is first of the simplest and most complete and robust algebraic structure, on which to base cryptography design. Groups which obey commutative or symmetric property are known as Abelian groups.

It was observed that for a non-group say, $y = xa$, which is not limited (not closed), but over infinite real numbers, or integers, it is easy for an intruder over time to map, or guess, the exponential pattern, from the random samples eavesdropped. So it was modified to $y = xa \bmod (n)$ [**1-3**], where *a*, *x*, *y*, *n* are integers, then the input-output relationship, (originally, an exponential relationship), between *x*, and *y* values now becomes more random, and hence it becomes much harder for an intruder to guess any pattern, [4]. At the same time, given *y* and *n* publicly known values in public key cryptography, it becomes very difficult to guess *x*. This is due to the hardness of the discrete log problem which is due to the group closure requirements, and is achieved via the trapdoor, modulo (n) function [**5-7**].

The mathematics of elliptic curves, used in cryptography, uses the fundamental basic theory as above, and is also based on diophantine equations. The elliptic curve used as the underlying field, (EC) $y^2 = x^3 + ax + b$, all variables, *x*, *y*, and parameters, *a*, *b* must be integers. EC is used in Galois Field GF(p), where p is a prime number as typically modulo arithmetic, or characteristic 2 fields, such as irreducible polynomial fields, eg: $GF(2^m)$ as the latter are easier to implement by shift, and xor circuits [**4,8**] . ECC is now an accepted standard, ANSI X9.42, Public Key Cryptography Systems, X.509.

The remainder of this paper is organized as follows. A brief introduction on Algebraic structure and finite fields is provided in section 2 and 3, respectively followed by elliptic curve operations over finite field and point representation in elliptic curve. The operations in these sections are defined on affine coordinate system. Section 6 provides the Group law required in elliptic curve cryptosystems to achieve security. ECDH key exchange algorithm presented in section 7 illustrates the use of elliptic curve over finite field.

2. ALGEBRAIC STRUCTURE
A non empty set G equipped with one or more binary operations is called an algebraic structure. Let (G, *) be an algebraic structure. And let G satisfies following properties:

(a) G is closed w.r.t *
(b) * is associative
(c) Existence of identity element
(d) Existence of inverse
(e) * is commutative
(f) Closure property w.r.t multiplication
(g) Associative w.r.t. multiplication
(h) Identity element w.r.t multiplication
(i) Inverse (multiplicative inverse)
(j) Commutative w.r.t multiplication
(k) Distributive.

Then Fig. 1 illustrates the different algebraic structure satisfying various properties through (a) – (k).

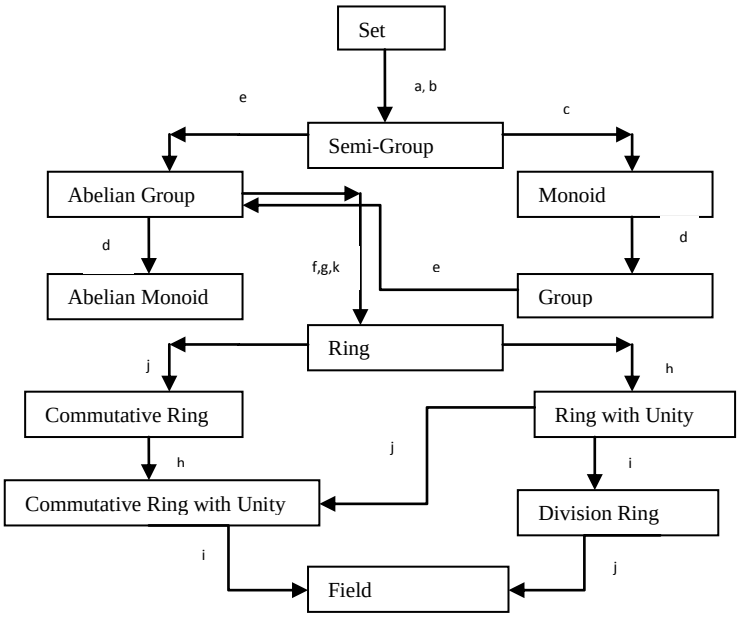

Fig 1: Heirarchy of algebraic structure

3. FINITE FIELD

Fields are abstractions of familiar number systems (such as the rational numbers Q, the real numbers R, and the complex numbers C) and their essential properties. They consist of a set F together with two operations, addition (denoted by +) and multiplication (denoted by ·), that satisfy the usual arithmetic properties:

(i) *(*F, +*)* is an abelian group with (additive) identity denoted by 0.

(ii) *(*F\ {0}, ·*)* is an abelian group with (multiplicative) identity denoted by 1.

(iii) The distributive law holds: $(a+b) \cdot c = a \cdot c + b \cdot c$ for all $a, b, c \in$ F. If the set F is finite, then the field is said to be *finite*.

3.1 Field operations

A field F is equipped with two operations, addition and multiplication. Subtraction of field elements is defined in terms of addition: for $a, b \in$ F, $a - b = a + (-b)$ where $-b$ is the unique element in F such that $b + (-b) = 0$ ($-b$ is called the *negative* of b). Similarly, division of field elements is defined in terms of multiplication: for $a, b \in$ F with $b = 0$, $a/b = a \cdot b^{-1}$ where b^{-1} is the unique element in F such that $b \cdot b^{-1} = 1$. (b^{-1} 1 is called the inverse of b.)

The order of a finite field is the number of elements in the field. There exists a finite field F of order q if and only if q is a prime power, i.e., $q = p^m$ where p is a prime number called the characteristic of F, and m is a positive integer. If $m = 1$, then F is called a prime field. If $m \geq 2$, then F is called an extension field. For any prime power q, there is essentially only one finite field of order q; informally, this means that any two finite fields of order q are structurally the same except that the labeling used to represent the field elements may be different. We say that any two finite fields of order q are isomorphic and denote such a field by F_q. Most Standards which specify the Elliptic curve Cryptographic techniques restrict the order of the underlying field to be an odd prime (q=p) or a power of 2($q=2^m$). [9-11]

3.2 Prime field Fp

Let p be a prime number. The integers modulo p, consisting of the integers {0, 1, 2, . . ., p −1} with addition and multiplication performed modulo p, is a finite field of order p. We shall denote this field by F_p and call p the *modulus* of Fp. For any integer a, a mod p shall denote the unique integer remainder r, $0 \leq r \leq p-1$, obtained upon dividing a by p; this operation is called reduction modulo p.

Example 1: (*prime field* F_{29}) The elements of F_{29} are {0, 1, 2, . . ., 28}. The following shows arithmetic operations in F_{29}.

- Addition: 17+20 = 8 since 37 mod 29 = 8.
- Subtraction: 17−20 = 26 since −3 mod 29 = 26.
- Multiplication: 17 · 20 = 21 since 340 mod 29 = 21.
- Inversion: 17−1 = 12 since 17 · 12 mod 29 = 1.

3.3 Binary Field F_2^m

Finite fields of order 2^m are called binary fields or characteristic-two finite fields. One way to construct $\mathbf{F_2^m}$ is to use a polynomial basis representation. Here, the elements of $\mathbf{F_2^m}$ are the binary polynomials (polynomials whose coefficients are in the field $F_2 = \{0, 1\}$) of degree at most m −1:

$$\mathbf{F_2^m} = \{a_{m-1} z^{m-1} + a_{m-2} z^{m-2} + \cdots + a_2 z^2 + a_1 z + a_0 : \ a_i = \{0, 1\}\}.$$

An irreducible binary polynomial $f(z)$ of degree m is chosen (such a polynomial exists for any m and can be efficiently found. Irreducibility of $f(z)$ means that $f(z)$ cannot be factored as a product of binary polynomials each of degree less than m. Addition of field elements is the usual addition of polynomials, with coefficient arithmetic performed modulo 2. Multiplication of field elements is performed modulo the reduction polynomial f (z). For any binary polynomial $a(z)$, $a(z)$ mod $f(z)$ shall denote the unique remainder polynomial $r(z)$ of degree less than m obtained upon long division of $a(z)$ by $f(z)$; this operation is called reduction modulo f (z).

Example 2: (*binary field* F_2^4) The elements of F_2^4 are the 16 binary polynomials of degree at most 3:

0	z^2 (0100)	z^3 (1000)	$z^3 + z^2$(1100)
1(0001)	$z^2 + 1$(0101)	$z^3 + 1$(1001)	$z^3 + z^2 + 1$(1101)
Z (0010)	$z^2 + z$*(0110)*	$z^3 + z$*(1010)*	$z^3 + z^2 + z$*(1110)*
$z+1$(0011)	$z^2 + z + 1$(0111)	$z^3 + z + 1$(1011)	$z^2 + z^2 + z + 1$.*(1111)*

The following shows arithmetic operations in F_2^4 with reduction polynomial $f(z) = z^4 + z + 1$. i.e. in binary form it is (10011)

- Addition: $(z^3 + z^2 + z) + (z^2 + z + 1) = z^3 + z$
- Subtraction: $(z^3 + z^2 + z) - (z^2 + z + 1) = z^3 + z$. (Note that since $-1 = 1$ in F_2, we have $-a = a$ for all $a \in \mathbf{F_2^m}$.)
- Multiplication: $(z^3 + z^2 + z) \cdot (z^2 + z + 1) = z^2 + 1$since $(z^3 + z^2 + z) \cdot (z^2 + z + 1) = z^5 + z + 1$. And $(z^5 + z + 1)$ mod $(z4 + z + 1) = z^2 + 1$.
- Inversion: $(z^3 + z^2 + z) - 1 = z^2$ since $(z3 + z2 + 1) \cdot z^2$ mod $(z4 + z + 1) = 1$.

3.4 Extension fields

The polynomial basis representation for binary fields can be generalized to all extension fields as follows. Let p be a prime and $m \geq 2$. Let $F_p[z]$ denote the set of all polynomials in the variable z with coefficients from F_p. Let $f(z)$, the reduction polynomial, be an irreducible polynomial of degree m in $F_p[z]$. Irreducibility of $f(z)$ means that $f(z)$ cannot be factored as a product of polynomials in Fp[z] each of degree less than m. The elements of F_p^m are the polynomials in $F_p[z]$ of degree at most m −1:

$$\mathbf{F_p^m} = \{a_{m-1} z^{m-1} + a_{m-2} z^{m-2} + \cdots + a_2 z^2 + a_1 z + a_0 : \ a_i \in F_p\}$$

Addition of field elements is the usual addition of polynomials, with coefficient arithmetic performed in F_p.

Multiplication of field elements is performed modulo the polynomial *f (z)*.

Example 3: Let p =251 and m =5. The polynomial $f(z) = z^5+z^4+12z^3+9z^2+7$ is irreducible in $F_{251}[z]$ and thus can serve as reduction polynomial for the construction of $F_{251}[z]$, the finite field of order 251^5. The elements of $F_{251}[z]$ are the polynomials in $F_{251}[z]$ of degree at most 4. The following are some examples of arithmetic operations in $F_{251}[z]$. Let $a = 123z^4+76z^2+7z+4$ and $b = 196z^4+12z^3+225z^2+76$.

- Addition: $a+b = 68z^4+12z^3+50z^2+7z+80$.
- Subtraction: $a-b = 178z^4+239z^3+102z^2+7z+179$.
- Multiplication: $a \cdot b = 117z^4+151z^3+117z^2+182z+217$.
- Inversion: $a^{-1} = 109z^4+111z^3+250z^2+98z+85$.

4. ELLIPTIC CURVE OVER FINITE FIELD

Elliptic Curve theory is an extension of group theory and Galois Field Theory. Most modulo operations are done, mod (number) or a modulo (prime number). They originate from Weierstrass equations. Cryptography on elliptic curves is based on scalar multiplication of points on the elliptic curves, as the basic operation. The location of the multiplicative inverse over the elliptic curve is the challenging part (as the factorization in RSA, discrete logarithm in Diffie-Hellman).
ECC operations involve arithmetic operations on an elliptic field, over a finite field. This is analogous to arithmetic operations over a ring of integers, or a modulo field, also known as Galois Field (GF). Operations over the real numbers are slow and inaccurate due to round-off error. Cryptographic operations need to be faster and accurate. To make operations on elliptic curve accurate and more efficient, the curve cryptography is defined over two finite fields.

- Prime field F_p and
- Binary field F_2^m

The field is chosen with finitely large number of points suited for cryptographic operations. Following section explains the EC operations on finite fields. The operations in these sections are defined on affine coordinate system in which each point is represented by the vector (x, y). Chapter 6 of Koblitz's book [9] provides an introduction to elliptic curves and elliptic curve systems. For a more detailed account, consult Menezes [12] or Blake, Seroussi and Smart [13]. Some advanced books on elliptic curves are Enge [14] and Silverman [15].

4.1 EC on Prime field F_p

The equation of the elliptic curve on a prime field F_p is $y^2 \bmod p = x^3 + ax + b \bmod p$, where $4a^2 + 23b^2 \bmod p \neq 0$. Here the elements of the finite field are integers between *0* and $p - 1$. All the operations such as addition, substation, division, multiplication involves integers between *0* and $p - 1$. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure. SEC specifies curves with p ranging between 112-521 bits [16, 17]. **T**he algebraic rules for point addition and point doubling are adapted for elliptic curves over F_p. The addition of two elliptic cuve points in F_p requires a few arithmetic operations (addition, subtraction, multiplication, inversion) in the underlying field

Point Addition

Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)
Let L = J + K where
L = (x_L, y_L), then $x_L = s^2 - x_J - x_K \bmod p$ (1)
$y_L = -y_J + s(xJ - xL) \bmod p$ (2)
$s = (y_J - y_K)/(x_J - x_K) \bmod p$, s is the slope of the line through J and K. (3)
If K = -J i.e. K = $(x_J, -y_J \bmod p)$ then J + K = O. where O is the point at infinity.
If K = J then J + K = 2J then point doubling equations are used.
Also J + K = K + J

Point Subtraction

Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)
Then J - K = J + (-K) where -K = $(x_K, -y_K \bmod p)$
Point subtraction is used in certain implementation of point multiplication such as NAF [18].

Point Doubling

Consider a point J such that J = (x_J, y_J), where $y_J \neq 0$
Let L = 2J
where L = (x_L, y_L), Then $xL = s^2 - 2x_J \bmod p$ (4)
$y_L = -y_J + s(x_J - x_L) \bmod p$ (5)
$s = (3x_J^2 + a) / (2y_J) \bmod p$, s is the tangent at point J and a is one of the parameters chosen with the elliptic curve . (6)
If $y_J = 0$ then $2_J = O$, where O is the point at infinity.

Example 4: Elliptic curve over the prime field F_{29}
Let p = 29, a = 4, and b = 20, and consider the elliptic curve $E: y^2 = x^3 + 4x + 20$ defined over F_{29}. Note that $=-16(4a^3 + 27b^2) = -176896 \neq 0 \pmod{29}$, so E is indeed an elliptic curve. The points in $E(F_{29})$ are the following:
∞ (2,6) (4,19) (8,10) (13,23) (16,2) 19,16) (27,2) (0,7) (2,23) (5,7) (8,19) (14,6) (16,27) (20,3) (27,27) (0,22) (3,1) (5,22) (10,4) (14,23) (17,10) (20,26) (1,5) (3,28) (6,12) (10,25) (15,2) (17,19) (24,7) (1,24) (4,10) (6,17) (13,6) (15,27) (19,13) (24,22)
Examples of elliptic curve addition and doubling are (5,22) + (16,27) = (13,6), and 2(5,22) = (14,6). Using equation (1)-(6)

4.2 Elliptic curve over binary field F_2^m

The equation of the elliptic curve on a binary field F_2^m is $y^2 + xy = x^3 + ax^2 + b$, where $b \neq 0$. Here the elements of the finite field are integers of length at most m bits. These numbers can be considered as a binary polynomial of degree $m - 1$. In binary polynomial the coefficients can only be 0 or 1. All the operation such as addition, substation, division, multiplication involves polynomials of degree $m - 1$ or lesser. The m is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure. SEC specifies curves with m ranging between 113-571 bits [18]. However, the algebraic rules for elliptic curves over F_2^m are same as elliptic curve on prime field

Point Addition

Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)
Let L = J + K where L = (x_L, y_L), then
$x_L = s^2 + s + x_J + x_K + a$ (7)
$y_L = s(x_J + x_L) + x_L + y_J$ (8)
$s = (y_J + y_K)/(x_J + x_K)$, s is the slope of the line through J and K. (9)

If K = -J i.e. K = (x_K, x_J + y_J) then J + K = O. where O is the point at infinity.
If K = J then J + K = 2J then point doubling equations are used.
Also J + K = K + J

Point Subtraction
Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)
Then J - K = J + (-K) where -K = (x_K, x_K + y_K)
Point subtraction is used in certain implementation of point multiplication such as NAF [18].

Point Doubling
Consider a point J such that J = (x_J, y_J), where $x_J \neq 0$
Let L = 2J where L = (x_L, y_L),
Then $x_L = s^2 + s + a$ (10)
$y_L = x_J^2 + (s + 1) * x_L$ (11)
$s = x_J + y_J / x_J$, *s* is the tangent at point J and *a* is one of the parameters chosen with the elliptic curve. (12)
If $x_J = 0$ then 2J = O, where O is the point at infinity

Example 5 : Consider the field F_2^4, defined by using polynomial representation with the irreducible polynomial $f(z)=z^4+z+1$. Then the element of F_2^4 are binary polynomial represented using powers of *g* , where g = (0010) is a generator for the field . The following are the powers of g:
g^0=(0001) g^1=(0010) g^2=(0100) g^3=(1000) g^4=(0011)
g^5=(0110) g^6=(1100) g^7=(1011) g^8=(0101) g^9=(1010)
g^{10}=(0111) g^{11}=(1110) g^{12}=(1111) g^{13}=(1101) g^{14}=(1001)
g^{15}=(0001)

In a true cryptographic application, the parameter *m* must be large enough to preclude the efficient generation powers of *g* otherwise the cryptosystem can be broken. In today's practice, *m* = 160 is a suitable choice.
The *g* notation allows the use of generator notation (g^e) rather than bit string notation, Also, using generator notation allows multiplication without reference to the irreducible polynomial $f(z) = z^4 + z + 1$. Example below illustrates this [19]

Consider the elliptic curve $y^2 + xy = x^3 + g^4x^2 + 1$. Here $a = g^4$ and $b = g^0 = 1$. The point (g^5,g^3) satisfies this equation over F_2^m : $y^2 + xy = x^3 + g^4x^2 + 1$
i.e. $(g^3)^2 + g^5 g^3 = (g^5)^3 + g^4 g^{10} + 1$
i.e. $g^6 + g^8 = g^{15} + g^{14} + 1$
i.e. (1100) + (0101) = (0001) + (1001) + (0001)
i.e. (1001) = (1001)

4.3 Geometrical Definition of Point Addition and point Doubling

For any two points P(x1,y1)≠ Q(x2,y2) on an elliptic curve, EC group law point addition can be defined geometrically as: "If we draw a line through P and Q, this line will intersect the elliptic curve at a third point(-R). The reflection of this point about x-axis, R(x3,y3) is the addition of P and Q" .For P=Q , point doubling, Geometrically if we draw a tangent line at point P, this line intersects elliptic curve at point a point (-R). Then, R is the reflection of this point about x-axis. This chord-tangent-rule for point addition and doubling is illustrated in figure 2 and 3 respectively. Figure 4 represents the concept of inverse element in finite field. Neutral element O is a point with y = ∞, which is added to the curve and Inverse element -P is the symmetric point of P

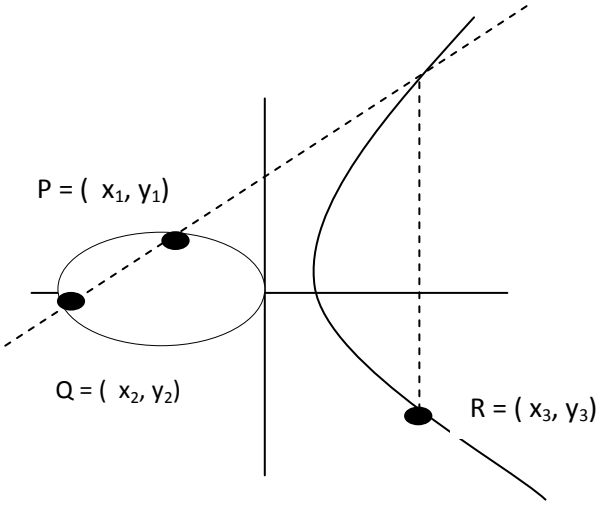

Fig 2: Point Addition P+Q =R

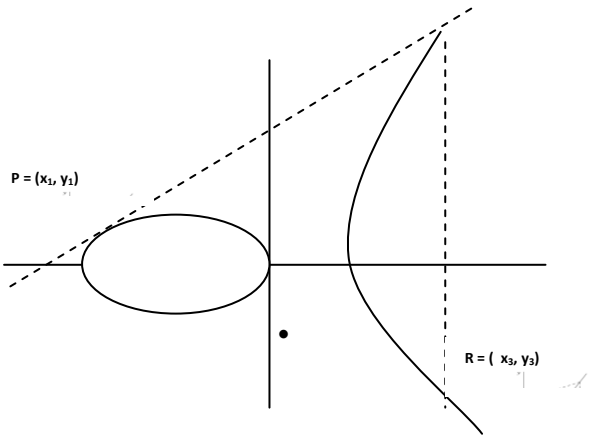

Fig 3: Point Doubling P +P = 2P =R

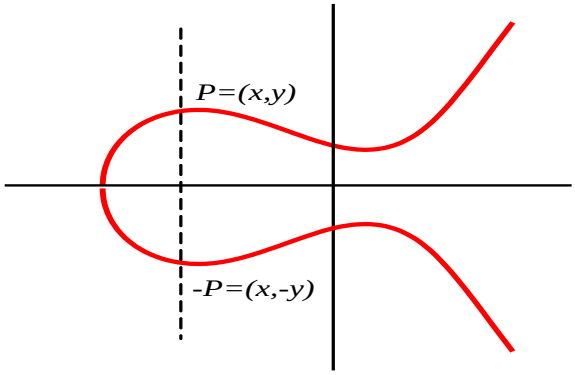

Fig 4 : Inverse element -P

5. ELLIPTIC CURVE POINT REPRESENTATION

The elliptic curve points can be represented in the following coordinate systems :(Affine (A), Projective (P), Jacobian (J), Chudnovsky brothers Jacobian (J C), and Modified Jacobian (JM). Each coordinate system requires different number of operations to perform point addition and doubling, and therefore different execution times. Also each coordinate system differs in the number of underlying finite field elements used to represent an elliptic curve point. This determines the capacity for storing or transferring elliptic curve points. Table 1 shows the representation of elliptic curve points as well as number of GF(p)elements used for storing a point in any of the five coordinate system. Table 2 shows the number of operations needed to compute elliptic curve point addition or doubling [20]. Here M is multiplication; S is Squaring and I in inversion.

Table 1 Representation of point and number of GF (p) elements

Coordinate system	Coordinates	Elements in GF (p)
Affine *A*	(x, y)	2
Projective *P*	(X, Y, Z)	3
Jacobian *J*	(X, Y, Z)	3
Chudnovsky Jacobian J^c	(X,Y,Z,Z^2, Z^3)	5
Modified Jacobian J^m	(X, Y, Z, Z^4)	4

Table 2 : Number of operations for adding and doubling points in different coordinate system.

Coordinate system	Addition	Doubling
Affine A	2M + S + I	2M + 2S + I
Projective P	12M + 2S	8M + 5S
Jacobian J	12M + 4S	4M + 6S
Chudnovsky Jacobian JC	11M + 3S	5M + 6S
Modified Jacobian J M	13M + 6S	4M + 4S

It is possible to mix different coordinate systems, i.e. to add two points where one is represented in some coordinate system, and the other point is represented in another coordinate system. Since operations in coordinate systems differ in their computational, mixing of coordinate systems is efficient [21]. When mixing different coordinate systems, some computational overhead is added because of the conversions between coordinates. Table 3 estimates the computational timings [20]

Table 3 Number of operations for conversion between coordinate systems.

From/To	Affine A	Projective P	Jacobian J	Chudnovsky Jacobian JC	Modified Jacobian J M
Affine A	0	2M	3M+S	3M+S	4M+2S
Projective P	2M+ I	0	2M+S	3M+S	3M+2S
Jacobian J	3M+S+I	M+S+I	0	M+S	M+2S
Chudnovsky Jacobian JC	3M+S+I	M+S+I	0	0	M+S
Modified Jacobian J M	3M+S+I	M+S+I	0	M+S	0

6. BASIC CONCEPT

6.1 Group law

Let *E* be an elliptic curve defined over the field *K*. There is a chord-and-tangent rule for adding two points in *E (K)* to give a third point in *E (K)*. Together with this addition operation, the set of points *E (K)* forms an abelian group with ∞ serving as its identity. It is this group that is used in the construction of elliptic curve cryptographic systems.

Let *E* be an elliptic curve defined over F*q*. The number of points in *E*(F*q*), denoted #*E*(F*q*), is called the order of *E* over F*q* . Since the Weierstrass equation has at most two solutions for each $x \in Fq$, $\#E(Fq) \in [1, 2q+1]$ provided by Hasse theorem. It states that number of points satisfying the elliptic curve falls in the range $q + 1 - \sqrt{q} \leq \#E(Fq) \leq q + 1 + 2\sqrt{q}$.

6.2 Admissible orders of elliptic curves

Let $q = p^m$ where *p* is the characteristic of F*q* . There exists an elliptic curve *E* defined over F*q* with $\#E(Fq) = q + 1 - t$ where *t* is the trace of *E*, if and only if one of the following conditions holds:

- (i) $t \equiv 0 \pmod p$ and $t^2 \leq 4q$.
- (ii) *m* is odd and either
  - (a) $t = 0$; or
  - (b) $t^2 = 2q$ and $p = 2$; or
  - (c) $t^2 = 3q$ and $p = 3$.
- (iii) *m* is even and either
  - (a) $t^2 = 4q$; or
  - (b) $t^2 = q$ and $p \equiv 1 \pmod 3$; or
  - (c) $t = 0$ and $p \equiv 1 \pmod 4$.

Hence for any prime *p* and integer *t* satisfying $|t| \leq 2\sqrt{p}$, there exists an elliptic curve *E* over F*p* with $\#E(Fp) = p + 1 - t$. In other words the order of elliptic curve (F*p*) is roughly equal to size p in the underlying field [22]. This is illustrated in following example along with Table 4.

Example 6: Let $p = 37$. Table 2 lists, for each integer *n* in the Hasse interval $[37+1-2\sqrt{37}, 37+1+2\sqrt{37}]$, the coefficients *(a, b)* of an elliptic curve $E: y^2 = x^3 + ax + b$ defined over F_{37} with

#E (F_{37}*)* = *n*. The order *#E* (F*q*) can be used to define super singularity of an elliptic curve.

Table 4. Admissible orders n = #*E* (F_{37}*) of elliptic curves E: $y^2 = x^3 + ax + b$

n	(a,b)	n	(a,b)	n	(a,b)	n	(a,b)	n	(a,b)
26	5,0	31	2,8	36	1,0	41	1,16	46	1,11
27	0,9	32	3,6	37	0,5	42	1,9	47	3,15
28	0,6	33	1,13	38	1,5	43	2,9	48	0,1
29	1,12	34	1,18	39	0,3	44	1,7	49	0,2
30	2,2	35	1,8	40	1,2	45	2,14	50	2,0

6.3 Group structure of an Elliptic curve

Let Z_n to denote a cyclic group of order *n* and *E(*F*q)* be an abelian group of rank 1 or 2. Then *E (*F*q)* is isomorphic to Z*n*1© Z*n*2 where *n*1 and *n*2 are uniquely determined positive integers such that *n*2 divides both *n*1 and *q* −1. Also *#E (*F*q)* = *n*1*n*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. If *n*2 is a small integer (e.g., *n* = 2, 3 or 4), then *E(*F*q)* is almost cyclic. Since *n*2 divides both *n*1 and *q* −1, one expects that *E (*F*q)* is cyclic or almost cyclic for most elliptic curves *E* over F*q* . The following two example illustrates the group structure over prime field and binary field.

Example7: *E* : $y^2 = x^3 + x + 1$ defined over F_{11}. Since 11 is prime, *E (*F_{11}*)* is a cyclic group and any point in *E(*F_{11}*)* except for ∞ is a generator of *E(*F_{11}*)*. The following shows that the multiples of the point *P = (*1, 5*)* generate all the points in *E(*F_{11}*)*. So the order of *P = (*1,5*)* is 13 that is the total number of coordinates or elements of group.[20]

0P= ∞ 1P=(1,5) 2P=(3,3) 3P=(8,2)
4P= (6,5) 5P= (4,6) 6P= (0,10) 7P= (2,0)
8P= (0,1) 9P=(4,5) 10P=(6,6) 11P=(8,9)
12P= (3, 8) 13P= (1, 6)

Example8: Consider F_{2^4} as represented by the reduction polynomial f (z) =z^4+z+1. The elliptic curve *E:* $y^2 + xy = x^3 + g^3 x^2 + (g^3 + 1)$ defined over F_{2^4} has *#E(*F_{2^4}*)* = 22 . Since 22 does not have any repeated factors, *E(*F_{2^4}*)* is cyclic. The point *P = (*g^3, 1*) = (*1000,0001*)* has order 11; its multiples are shown below.[23]

0*P* =∞ 1*P* = *(*1000, 0001*)* 2*P* = *(*1001, 1111*)*
3*P* = *(*1100, 0000*)* 4*P* = *(*1111, 1011*)* 5*P* = *(*1011, 0010*)*
6*P* = *(*1011, 1001*)* 7*P* = *(*1111, 0100*)* 8*P* = *(*1100, 1100*)*
9*P* = *(*1001, 0110*)* 10*P* = *(*1000, 1001*)*

7. ELLIPTIC CURVE DIFFIE HELMAN PROTOCOL

ECDH, a variant of DH, is a key agreement algorithm. For generating a shared secret between A and B using ECDH, both have to agree upon Elliptic Curve domain parameters. An overview of ECDH is given below.

7.1 Key Agreement Algorithm

For establishing shared secret between two device A and B

E1. Let d_A and d_B be the private key of device A and B respectively, Private keys are random number less than n, where n is a domain parameter.
E2. Let $Q_A = d_A * G$ and $Q_B = d_B * G$ be the public key of device A and B respectively, G is a domain parameter
E3. A and B exchanged their public keys
E4. The end A computes $K = (x_K, y_K) = d_A * Q_B$
E5. The end B computes $L = (x_L, y_L) = d_B * Q_A$
E6. Since K=L, shared secret is chosen as x_K

7.2 ECDH - Mathematical Explanation

To prove the agreed shared secret K and L at both devices A and B are same, from E2, E4 and E5

$K = d_A * Q_B = d_A * (d_B * G) = (d_B * d_A) * G = d_B * (d_A * G) = d_B * Q_A = L$

Hence K = L, therefore $x_K = x_L$

Since it is practically impossible to find the private key d_A or d_B from the public key Q_A or Q_B, it is not possible to obtain the shared secret for a third party.

7.3 Relating with finite field

Here

- Private Key d_A and d_B are scalar quantity.
- *n* is prime number.
- Elliptic curve E, a domain parameter, satisfies the cyclic abelian property.
- G, a domain parameter, is the generator point of elliptic curve E on which both devices have agreed upon.
- Q_A and Q_B public key of device A and B respectively are coordinate points satisfying the elliptic curve E.
- Step E2 where $Q_A = d_A * G$ and $Q_B = d_B * G$, uses the concept of point addition and point doubling. i.e. $Q_A = \sum_{d_A\ times} G$ and $Q_B = \sum_{d_B\ times} G$
- To verify K = L , algorithm uses commutative property of Abelian group.

8. CONCLUSION

(a) The study concludes that Abelian cyclic groups , that are defined over finite fields and have desirable properties concerning their orders and their associated pairings, are extensively used in cryptography, as the order of the sender-receiver transmission does not confuse the common key. Also

(b) The abelian group of points of an elliptic curve due to the smaller key size, maintains the same level of security as in conventional cryptosystems. Shorter key sizes make Elliptic suitable for lightweight computing, bandwidth, power devices as mobiles, laptops, mobile web browsers etc.

(c) In the case of elliptic curves, we found the operation "+" to be compatible with its geometry, and later, a group structure. When '+' operation evaluated, to provide evidence for abelian group law an identity element, inverse elements, abelian properties, and associability were clarified [24]

(d) Since modular arithmetic involves no floating-point operations, the mathematical calculations are more accurate and efficient than the real number arithmetic. The modulo (n) operation causes the domain to have finite number of

members. This ensures the problem is solvable for the valid receiver, as well as for the problem to be hard e.g. discrete log for Diffie-Hellman or Elliptic curves and prime factorization for RSA.

(e) Points of the elliptic curve can be represented in different coordinate system depending upon the application. But since operations in coordinate system differ in their computational timings, it is advantageous to mix different coordinate system. And in situations where inversion is F_p/F_2^m is expensive relative to multiplication, it may be advantages to represent points using projective coordinates.

(f)This suggests that ECCs are superior to currently deployed public key cryptosystems since not only do they offer a greater level of security when the underlying parameters are chosen correctly, but they offer a greater advantage due to its shorter key sizes, faster generation of systems, smaller space requirements and efficient implementation techniques.
The study can be further extended general class of curves , Hyper Elliptic Curves.

9. ACKNOWLEDGEMENT

We would like to express our gratitude to all those who gave us the possibility to complete this report. We are deeply indebted to Prof. Sanjay Sharma, Associate Professor in Mathematics Dept. and Prof M.K. Kowar, Professor in ETC Dept. whose help, stimulating suggestions, knowledge, experience and encouragement helped us in all the times of study and analysis of the paper in the pre-research period.

10. REFERENCES

[1] Murphy T, "Course 373-Finite Fields" , University of Dublin, Trinity College School of Mathematicshttp://www.maths.tcd.ie/pub/Maths/ Courseware /FiniteFields/FiniteFields.pdf, 2006.

[2] Schneier, Bruce, "Applied Cryptography: Protocols, Algorithms and Source Code in C" , John-Wiley and Sons, New York, 1994. ISBN: 0-471-5975602, 1994.

[3] Stinson, Douglas R, "Cryptography: Theory and Practice" , CRC Press, Boca Raton, Florida, 1995, ISBN: 0-8493-8521-0, 1995.

[4] Certicom : "ECC Tutorial" , http://www.certicom. com/index.php?action=ecc,ecc_tutorial , 2006.

[5] Goldreich, Oded , " Foundations of Cryptography" , Cambridge University Press, Cambridge 2001, ISBN 0-521-79172-3, 2001.

[6] Mollin, Richard , "Introduction to Cryptography", Chapman & Hall/CRC, Boca Raton, 2000, ISBN" 1-58488-127-5, 2005.

[7] Welsh, Dominic ,"Codes and Cryptography", Oxford University Press, Oxford. 1990, ISBN 0-19-853287-3, 1990

[8] Balasubramanian, R. , "Elliptic Curves and Cryptography", in Bhandari A.K., Nagraj D.S., Ramakrishnan, B., Venkataraman T.N., (editors), Elliptic Curves, Modular Forms, and Cryptography, Hindustan Book Agency, New Delhi, 2003. ISBN 81-85931-42-9, pp 325-345, 2003.

[9] N. Koblitz, "A Course in Number Theory and Cryptography" , 2nd edition, Springer-Verlag, 1994.

[10] R. McEliece, "Finite Fields for Computer Scientists and Engineers" , Kluwer Academic Publishers, Boston,1987.

[11] R. Lidl and H. Niederreitter, "Introduction to Finite Fields and their Applications", Cambridge UniversityPress , 1984

[12] A. Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, Boston , 1993

[13] I. Blake, G. Seroussi and N. Smart, "Elliptic Curves in Cryptography" , Cambridge University Press, 1999.

[14] A. Enge, "Elliptic Curves and Their Applications to Cryptography—An Introduction" , Kluwer Academic Publishers, 1999.

[15] J. Silverman, "The Arithmetic of Elliptic Curves" , Springer-Verlag, 1986.

[16] Certicom, Standards for Efficient Cryptography, "SEC 1: Elliptic Curve Cryptography" , Version 1.0, September 2000, Available at http://www.secg.org /download/aid385/sec1_final.pdf

[17] Certicom, Standards for Efficient Cryptography, "SEC 2: Recommended Elliptic Curve Domain Parameters" , Version 1.0, September 2000, Available at http://www.secg.org/download/aid-386/sec2_final.pdf

[18] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, "Software Implementation of Elliptic Curve Cryptography over Binary Fields" , 2000, Available at http://citeseer.ist.psu.edu/hankerson00software.html

[19]http://www.certicom.com/index.php/41-an-example-of-an- elliptic-curve-group-over-f2m

[20] Michal Sramka, Otokar Grosek, " Efficiency of the Elliptic Curve Cryptosystems", AMS Subject Classification: 11T71, 11G07, work was supported by VEGA grant 1/7611/20.

[21]COHEN, H., - Miyaji, A.,- Ono, T, " Efficient Elliptic Curve Exponentiation Using Mixed Coordinates", Advances in Cryptology – ASIACRYPT 98, LNCS 1514, Springer-Verlag, (1998), pp. 51-65.

[22] W. WATERHOUSE., "Abelian varieties over finite fields. Annales Scientifiques" , de l'E´cole Normale Superieure, 4e Serie, 2:521–560, 1969.

[23] Darell Hankerson, Alfred Menezees and Scott Vanston, " Guide to Elliptic Curve Cryptography" Springer- Verlag, New York, Inc. 2004

[24] Dipti Aglawe , Samta Gajbhiye , "Software Implementation of Cyclic Abelian Elliptic Curve using Matlab", International Journal of Computer Applications (0975 – 8887) Vol 42(6), p: 43-48 March 2012