JOURNAL OF COMPUTING TECHNOLOGIES

ISSN 2278 – 3814

Available online at www.jctjournals.com

Volume 1 Issue 2, June 2012

Performance Evaluation of ECC

[1]G. Jai Arul Jose, [2]Dr. C. Suyambulingom

[1]*Research Scholar, Sathyabama University, Chennai, INDIA*

[2]*Professor, Dept. of Mathematics, Tamil Nadu Agricultural University, Coimbator, INDIA*

Abstract— **to date Elliptic Curve Cryptography is gaining wide acceptance as an alternative to the conventional cryptosystems (DES, RSA, AES, etc.) which tend to be power hungry. Elliptic Curve ciphers require less computational power, memory and communication bandwidth giving it a clear edge over the traditional Crypto-Algorithms. This paper describes the basic mathematical background of Elliptic Curve Cryptography (ECC), and briefly analysis its strength.**

Keywords—**ECC, RSA, Cryptography, Security, Cryptanalysis**

1. INTRODUCTION

Cryptographic mechanisms based on Elliptic curves depend on arithmetic involving points of the curve. Elliptic curve cryptography shortly known as ECC is a public key cryptography. A full understanding of this needs a good background of mathematics. In this we initially give the mathematical concepts necessary to understand and implement the arithmetic operations.

2. FINITE FIELDS

A finite field consists of a finite set of objects called field elements together with the description of two operations - addition and multiplication-that can be performed on pairs of finite elements. These operations must possess certain properties.

It turns out that there is a finite field containing q field elements if and only if q is a power of a prime number, and furthermore that in fact for each such q there is precisely one finite field. The finite field containing q elements is denoted by F_q.

Here only two types of finite fields F_q are used - finite fields F_p with q = p, p, an odd prime which are called prime finite fields, and finite fields F_{2^m} with $q = 2^m$ for some $m \geq 1$ which are called characteristic 2 finite fields. In order to precisely specify cryptographic schemes based on ECC the details about the above fields are essential and hence given below:

2.1. THE FINITE FIELD F

The finite field F_p is the prime field containing p elements. Although there is only one prime finite field F_p on each odd prime p, there are many different ways to represent the elements of F_p. Here the elements of F_p are represented by the set of integers: {0, 1, 2, ..., p-1} with addition and multiplication defined as follows:

(i) Addition: If a, b $\in F_p$, then a + b = r in F_p where r $\in$[0, p-1] is the remainder when the integer (a + b) is divided by p. This is known as addition modulo p and written as a + b ≡ r (mod p)

E.g.: if p = 7, a = 4, b = 5 i.e. 4 and 5 $\in$ [0, 7]

Now a + b = 4 + 5 = 9 = 2 since 9 = 1 x 7 + 2 and multiple of 7 is removed.

∴ 4 + 5 = 2 in F_7

(ii) Multiplication: If a, b $\in F_p$, then a. b = s in F_p, where s $\in$[0, p-1] is the remainder when ab is divided by p. This is known as multiplication modulo p and written as a.b = s (mod p)

i.e. ab - s is a multiple of p.

in the example 4.5 = 20 = 6 (mod 7)

= -1(mod 7)

Here 0 is the addition identity and 1 is the multiplication identity. For subtraction and division we need the definition of inverse in the field.

(iii) Additive inverse: The additive inverse of a $\in F_p$ is (-a) or it is the unique solution of the equation $a + x \equiv 0 \pmod p$.

(iv) Multiplicative inverse: For $a \in F_p$, the multiplicative inverse is a^{-1} which is also the unique solution of $ax \equiv 1 \pmod p$.

2.2. FINITE FIELD F_{2^m}

The finite field F_{2^m} is the characteristic 2 finite field containing 2^m elements. Although there is only one characteristic 2 finite field F_{2^m} for each power 2^m of 2 with $m \geq 1$, there are many different ways to represent the elements of F_{2^m}.

The elements of F_{2^m} should be represented by the set of binary polynomials of degree m - 1 or less:

$\{a_0 + a_1x + \ldots + a_{m-1}x^{m-1} : a_i \in \{0,1\}\}$

Here addition and multiplication are in terms of the binary polynomial f(x) of degree m, known as the reduction polynomial.

(i) Addition: If $a = a_0 + a_1x + \ldots + a_{m-1}x^{m-1}$, $b = b_0 + b_1x + \ldots + b_{m-1}x^{m-1}$ both $\in F_{2^m}$, then $a + b = r$ in F_{2^m} where $r = r_0 + r_1x + \ldots + r_{m-1}x^{m-1}$ where $r_i = a_i + b_i \pmod 2$.

(ii) Multiplication: If a and b are as defined above, $a.b = s$ in F_{2^m}, Where $s = s_0 + s_1x + \ldots + s_{m-1}x^{m-1}$ is the remainder when ab is divided by f(x) with all the coefficient arithmetic performed modulo 2.

As in the 2.1 the addition identify is the polynomial 0 and the multiplicative identify is the polynomial 1.

(iii) Additive inverse: As in the previous case the solution of $a + x = 0$ in F_{2^m}.

(iv) Multiplicative inverse: It is the solution of $ax = 1$ in F_{2^m}. The characteristic 2 finite field F_{2^m} should have:

$m \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}$

and all the algebraic operations should be performed by using the irreducible binary polynomial of degree m given in Table 1.

Examples

(i) The elements of F_{29} are [0, 1, …, 28]

Addition: 17+20 = 8 since 37 mod 29 = 8

Subtraction: 17-20=26 since -3mod29=26

Multiplication: 17.20 = 21 since 340 mod 29 = 21

Inversion: $17^{-1}=12$ since 17.12mod29=1

(ii) The elements of $F_2{}^4$ are the 16 binary polynomials of degree atmost 3:

0, 1, z, z+1, z^2, z^2+1, z^2+z, z^2+z+1, z^3, z^3+1, z^3+z, z^3+z+1, z^3+z^2, z^3+z^2+1, z^3+z^2+z, z^3+z^2+z+1.

TABLE 1

Field	Reduction polynomial f(x)
$F_2{}^{113}$	$x^{113} + x^9 + 1$
$F_2{}^{131}$	$x^{131} + x^8 + x^3 + x^2 + 1$
$F_2{}^{163}$	$x^{163} + x^7 + x^6 + x^3 + 1$
$F_2{}^{193}$	$x^{193} + x^{15} + 1$
$F_2{}^{233}$	$x^{233} + x^{74} + 1$
$F_2{}^{239}$	$x^{239} + x^{36} + 1$ or $x^{239} + x^{158} + 1$
$F_2{}^{283}$	$x^{283} + x^{12} + x^7 + x^5 + 1$
$F_2{}^{409}$	$x^{409} + x^{87} + 1$
$F_2{}^{571}$	$x^{571} + x^{10} + x^5 + x^2 + 1$

Arithmetic operation on $F_2{}^4$ with the reduction polynomial $f(z) = z^4 + z+1$ is

Addition: $(z^3 + z^2+1) + (z^2 + z+1) = z^3+z$

Subtraction: $(z^3 + z^2+1) - (z^2 + z+1) = z^3+z$ [$\therefore$ -1 = 1 in F_2]

Multiplication: $(z^3 + z^2+1) . (z^2 + z+1) = z^2+1$

i.e. $(z^3 + z^2+1) + (z^2 + z+1) = z^5 + 2(z^4 + z^3 + z^2) + z + 1 = z^5 + z + 1$ [$\therefore z = 0$ in F_2]

$(z^5 + z + 1) \bmod (z^4 + z + 1) = z^2 + 1$

Inversion: $(z^3+z^2+1)^{-1}=z^2$ since $(z^3+z^2+1)z^2 \bmod (z^4 +z + 1) = 1$.

In F_2 the arithmetic operations are made simple if binary representation is made.

Let $A = z^3 + z^2 + 1$ has binary representation

		$1\ 1\ 0\ 1_2$
$B = z^2 + z + 1$		$0\ 1\ 1\ 1_2$
	Sum A+B =	$1\ 0\ 1\ 0_2$
	A-B =	$1\ 0\ 1\ 0_2$
	A*B =	$0\ 1\ 0\ 1_2$
	A/B =	$0\ 1\ 0\ 0_2$

Thus polynomial arithmetic can be made simple though binary representation in F_2.

3. ELLIPTIC CURVES

An elliptic curve over F_q is defined in terms of the solutions to an equation in F_q. The form of the equation defining an elliptic curve over F_q differs depending on whether the filed is a prime finite field or a characteristic 2 finite field.

3.1. ELLIPTIC CURVES OVER F_p

Let F_p be a prime finite field so that p is an odd prime number and let a, b $\in F_p$ satisfy $4a^3 + 27b^2 \equiv 0 \pmod p$. Now an elliptic curve $E(F_p)$ defined by the parameters a, b $\in F_p$ consists of the set of solutions or points P = (x, y) for x, y $\in F_p$ to the equation

$y^2 = x^3 + ax + b \pmod p$ -------- (i)

together with an extra point 0 called the point at infinity. (i) is called the defining equation of $E(F_p)$. For a given point $P = (x_p, y_p)$, x_p is called the x - coordinate of P and Y_P; the y-coordinate of P. The number of points is denoted by $\# E(F_p)$ and it satisfies the inequality.

$p + 1 - 2\sqrt{p} \leq \# E(F_p) \leq p + 1 + 2\sqrt{p}$

Point Addition Rule

1. Rule to add the point at infinity to it self
$0 + 0 = 0$.
2. Rule to add point at infinity to any other point
$(x, y) + 0 = 0 + (x, y) = (x, y)$ for all $(x, y) \in E(F_p)$
3. Rule to add two points with the same x - coordinates
$(x, y) + (x, -y) = 0$ for all $(x, y) \in E(F_p)$
i.e. the negative of (x, y) is $-(x, y) = (x, -y)$
4. Rule to add two points with different x-coordinates:
Let (x_1, y_1) and (x_2, y_2) are two points in $E(F_p)$ and that $x_1 \neq x_2$,
Then $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$; where

$$x_3 = \lambda^2 - x_1 - x_2 \pmod p,$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p,$$
$$\text{where } \lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$

5. Doubling or Rule to add a point to itself,
Let $(x_1, y_1) \in E(F_p)$ with $y_1 \neq 0$.
Then $(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$ where
$x_3 = \lambda^2 - 2x_1 \pmod p$,
$y_3 = \lambda (x_1 - x_3) - y_1 \pmod p$, and
$\lambda = \frac{3x_1^2 + a}{2y_1} \pmod p$
6. Point multiplication:
In point multiplication a point P on the elliptic curve is multiplied with a scalar K using the elliptic curve equation to obtain another point Q on the same elliptic curve.
i.e. $KP = Q$
Suppose $K = 23$
Then $KP = 23P = 2\,(2(2(2P)+P)+P)+ P$.
Thus point multiplication uses point addition and point doubling repeatedly to find the result.

3.2. ELLIPTIC CURVES OVER F_{2^m}

Let F_{2^m} be a characteristic 2 finite field, let a, $b \in F_{2^m}$ such that $b \neq o$ in F_{2^m}. Then a (non-super singular) elliptic curve $E(F_{2^m})$ over F_{2^m} defined by the parameters a, $b \in F_{2^m}$ consists of the set of solutions or points $P = (x, y)$ for $x, y \in F_{2^m}$ to the equation.

$$y^2 + xy = x^3 + ax^2 + b \text{ in } F_{2^m}$$

together with an extra point 0 called the point at infinity.The number of points on $E(F_{2^m})$ denoted by $\#E(F_{2^m})$ satisfies the in equality

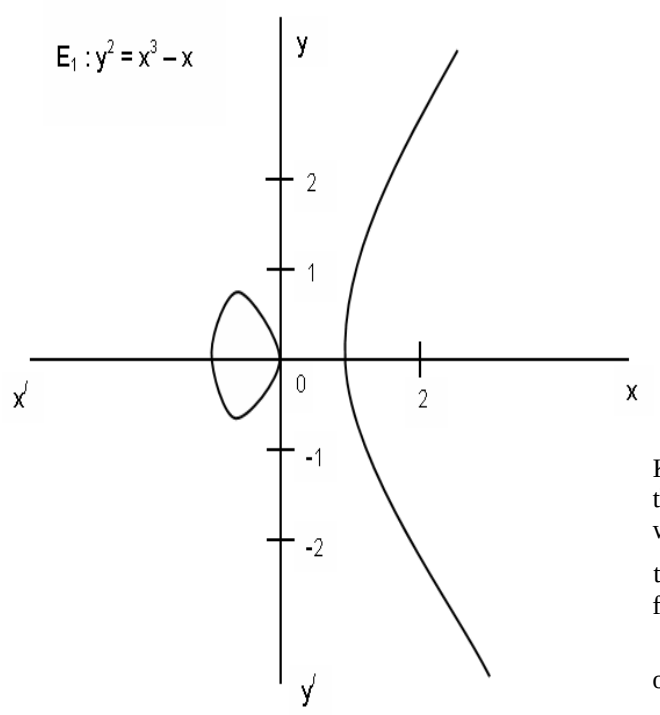
$$2^m+1 - 2\sqrt{2^m} \leq \#E(F_{2^m}) \leq 2^m +1 + 2\sqrt{2^m}.$$

The addition rule here are
1. Rule to add the point at infinity to itself:
$0 + 0 = 0$.
2. Rule to add point at infinity to any other point:
$(x, y) + 0 = 0 + (x, y) = (x, y)$ for all $(x, y) \in E(F_p)$
3. Rule to add two points with the same x – coordinates:
$(x, y) + (x, x + y) = 0$ for all $(x, y) \in E(F_p)$
i.e. the negative of $(x, y) = -(x, y) = (x, x + y)$
4. Rule to add two points with different x-coordinates:
Let (x_1, y_1) and $(x_2, y_2) \in E(F_2^m)$ be two points such that $x_1 \neq x_2$, Then $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$;
Where $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$ in F_2^m
$y_3 = \lambda (x_1 + x_3) + x_3 + y_1$ in F_2^m,
and $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$ in F_{2^m}
5. Rule to add a point to itself (doubling),
Let $(x_1, y_1) \in E(F_{2^m})$ be a point with $x_1 \neq 0$.
Then $(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$ where
$x_3 = \lambda^2 + \lambda + a$ in F_{2^m}
$y_3 = x_1^2 + (\lambda + 1)\, x_3$ in F_{2^m} and
$\lambda = x_1 + \frac{y_1}{x_1}$ in F_{2^m}
As before given a point P the scalar multiplication is KP i.e. adding P repeatedly K times with itself.

4. GENERAL FORM OF EC EQUATION

An elliptic curve E over a field K is defined by the equation
$E = y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ -------(4.1)
where $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$, where Δ is the discriminant of E and in defined as follows
$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$
$d_2 = a_1^2 + 4a_2$
$d_4 = 2a_4 + a_1a_3$
$d_6 = a_3^2 + 4a_6$
$d_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2$.
If L is any extension field of K, then the set of L-rational Points on E is
$E(L) = \{ (x, y) \in L \times L : y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0 \} \cup \{ \infty \}$
where ∞ is the point at infinity.
Equation (4.1) is called a Weierstrass equation. The condition $\Delta \neq 0$ ensures that the elliptic curve is "smooth", that is there are no points at which the curve has two or more district tangent lines.
Example:
The equations
$E_1 = y^2 = x^3 - x$
$E_2 = y^2 = x^3 + \frac{1}{4}x + \frac{5}{4}$ are defined over the field R of real numbers.
They are graphed in figure 1 and 2.

Figure 1: Elliptic Curves over R



5. ELLIPTIC CURVE CRYPTOGRAPHY

To form a Cryptographic system using elliptic curves, it is necessary to find a ―hard problem‖ corresponding to factoring the product of two primes or taking the discrete logarithm.

Consider the equation Q = kP, where Q, P ∈ E_p(a, b) and k < p. It is relatively easy to calculate Q given k and P. but it is relatively hard to determine k given Q and P. This is called the discrete logarithm problem for elliptic curves.

Here is an example taken form the Certicom Web site (www.certicom.com). Consider the group E_{23}(9, 17). This is the group defined by the equation $y^2 \bmod 23 = (x^2 + 9x + 17) \bmod 23$. What is the discrete logarithm k of Q = (4, 5) to the base P = (16, 5)? The brute-force method is to compute multiples of P and Q is found. Thus

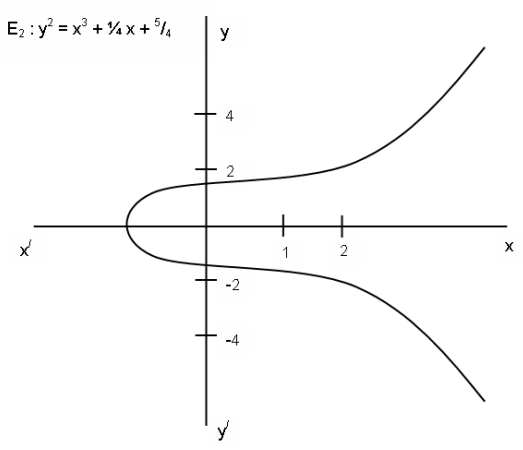
P = (16, 5); 2P = (20, 20); 3P = (14, 14); 4P = (19, 20);

5P = (13, 10); 6P = (7, 3); 7P = (8, 7);

8P = (12, 17); 9P = (4, 5)

Because 9P = (4, 5) = Q, the discrete logarithm is Q = (4, 5) to the base P = (16, 5). In a real application, k would be so large as to make the brute-force approach infeasible.

Figure 2: Elliptic Curves over R



5.1. KEY EXCHANGE

Key exchange using elliptic curves can be done in the following manner. First pick a large integer q, which is either a prime number p or an integer of the form 2^m and elliptic curve parameters a and b for the equation

$$y^2 \bmod p = (x^2 + ax + b) \bmod p$$

or for the equation

$$y^2 + xy = x^3 + ax^2 + b$$

This defines the elliptic group of points E_q(a, b). Next, pick a base point G = (x, y) in E_q(a, b) whose order is a very large value n. The order n of a point G on an elliptic curve is the smallest positive integer n such that nG = O. E_q(a, b) and G are parameters of the Cryptosystem known to all participants.

A key exchange between users A and B can be accomplishes as follows:

Global Public Elements

E_q(a, b) elliptic curve with parameters a, b and q, where q is a prime or an integer of the form 2^m

G Point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A $n_A < n$

Calculate public P_A $P_A = n_A \times G$

User B Key Generation

Select private n_B $n_B < n$

Calculate public P_B $P_B = n_B \times G$

Generation of Secret Key by User A

$K = n_A \times P_B$

Generation of Secret Key by User B

$K = n_B \times P_A$

1) A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the public key is a point in E_q(a, b).

2) B similarly selects a private key n_B and computes a public key $P_B = n_B \times G$
3) A generates the secret key $K = n_A \times P_B$. B generates the secret key $K = n_B \times P_A$.

To break this scheme, an attacker would need to be able to compute k given G and kG, which is assumed hard.

As an example provided by Ed Schaefer of Santa Clara University, take p = 211; $E_p(0, -4)$, which is equivalent to the curve $y^2 = x^3 - 4$; and G = (2, 2).

One can calculate that 240G = O.

A's private key is $n_A = 121$, A's public key is $P_A = 121(2, 2) = (115, 48)$. B's private key is $n_B = 203$, so B's public key is $P_B = 203(2, 2) = (130, 203)$. The shared secret key is $121(130, 203) = 203(115, 48)$.

The secret key is a pair of numbers. If this key is to be used as a session key for conventional encryption, then a single number must be generated.

5.2. ELLIPTIC CURVE ENCRYPTION/DECRYPTION

There are several approaches to encryption/decryption using elliptic curves. Here a simple approach is explained. The first task in this approach is to encode the plain text message m to be sent as an x-y point P_m. It is the point P_m that will be encrypted as a ciphertext and subsequently decrypted.

As with key exchange system, an encryption/decryption system requires a point G and an elliptic group $E_q(a, b)$ as parameters. Each user A selects a private key n_A and generates a public key

$P_A = n_A \times G$.

To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext C_m consisting of the pair of points

$$C_m = \{kG, P_m + kP_B\}$$

Note that A has used B's public key P_B. To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - {}_B(kG) = P_m$$

A has masked the message P_m by adding kP_B to it. Nobody but A knows the value of k, so even though P_B is a public key, nobody can remove the mask kP_B. However, A also includes a ―clue,‖ which is enough to remove the mask if one knows the private key n_B. For an attacker to recover the message, the attacker would have to compute k given G and kG, which is assumed hard.

As an example, taken from ―Koblitz, N., A Course in Number Theory and Cryptography‖, of the encryption, take p = 751; $E_p(-1, 188)$, which is equivalent to the curve $y^2 = x^3 - x + 188$; and G = (0, 376). Suppose that A wishes to send a message to B that is encoded in the elliptic point $P_m = (562, 201)$ and that A selects the random number k = 386. B's public key is $P_B = (201, 5)$. Then 386(0, 376) = (676, 558), and (562, 201) + 386(201, 5) = (385, 328). Thus A sends the cipher text {(676, 558), (385, 328)}.

6. THE SECURITY OF ELLIPTIC CURVE CRYPTOGRAPHY

The purpose of any Public-key Cryptosystem is to maintain the security and integrity of the resources, avoid the attack of any people, any event etc. while the anti-attack performance of the algorithm assures its security. In 6th International Cryptography Conferences in January 2000, Elliptic Curve Cryptography as well as RSA were the only two algorithms that were recommended. Actually in the term of security, Elliptic Curve Cryptography provides the highest strength per bit among all the Cryptosystems.

The security of Elliptic Curve Cryptography is depends on how difficult it is to determine k given kP and P. This is referred to as the elliptic curve logarithm problem. The fastest known technique for taking the elliptic curve logarithm is known as the Pollard Rho method.

TABLE 2: THE TIME SPENT ON CRACKING ALL SIZE KEYS

ECC Key (bit)	**RSA Key (bit)**	**Time**	**The number Compu-ters**	**Memor y**
112	430	<5 minute	105	Very small
160	760	600 months	4300	4GB
192	1020	3 million years	114	170GB
256	1620	10^{16} years	0.16	120TB

The longer the key is, the higher the security strength it has. But it adversely affects the running performance. In April 2000, there was a research on the cost of cracking the different encrypted algorithms, which was done by the RSA lab. Table 2 displays its results, where Elliptic Curve Cryptography shows its advantage in the condition of considering both running performance and strength. For the same security level, the key size of Elliptic Curve Cryptography is much shorter RSA's. In other words, Elliptic Curve Cryptography provides a more secure Cryptosystem for the same length as RSA. Table 2 also shows what the attacker need is the larger memory rather than the more computers with the increasing of key size. That means when Elliptic

Curve Cryptography is exploited, it needs smaller memory.

TABLE 3: KEY SIZE RATIO

ECC Key size (Bits)	**163**	**283**	**409**	**571**
Traditional Key Size (Bits)	1024	3072	7680	15360
Key Size Ratio	1 : 6	1 : 11	1 : 19	1 : 27

Table 3 is the key size ratio of Elliptic Curve Cryptography to traditional key with the strengthening of security, the Elliptic Curve Cryptography key size is smaller and it increases slowly. In general, Elliptic Curve Cryptography key is small, efficient and low power.

In addition to the above advantages, fast speed is another characteristic of Elliptic Curve Cryptography. As we know, RSA is based on large integer factorization; all the process is rather complicated and strict. Although Elliptic Curve Cryptography processes of creating private key is complicated, we can calculate Public-key very easily. Speed in the process of the decrypted and signature is rather faster. In the equivalent of security, the speed of exploiting 160-bit Elliptic Curve Cryptography is about 10 times faster than that of 1024-bit RSA or DSA.

7. PERFORMANCE ADVANTAGES OF ECC

In Table 4 rows 1 and 2 are taken from –V. Guptha, S. Guptha and S. Chang, Performance Analysis of Elliptic Curve Cryptography for SSL, ACM Wksp. Wireless Security, Mobicom 2002, Atlanta, GA, Sept. 2002, http://research.sun.com/prolects/crypto/performance.pdf‖, and do not claim to be optimized, but show two different platforms and are directly comparable to RSA numbers for the same platforms.

Rows 3 and 4 in Table 4 are taken from –M. Brown et al., Software implementation of the NIST Elliptic curves over prime fields, D. Naccache, Ed., Topics in Cryptology – CT-RSA 2001, LNCS, vol. 2020, Springer – Verlag, 2001, pp. 250 -65‖ and take advantage of the special form of the generalized Mersenne primes for the NIST curves given in –FIPS Pub 186 – 2, Digital Signature Standard (DSS), Jan.27, 2000. http://csrc.nist.gov/publications/fips/fips186–2/fips186-2-change1 .pdf‖ by using specialized routines for fast modular reduction for these primes –J. Solinas, ‖Generalized Mersenne Numbers,‖ Tech. rep.,1999, http:// www.cacr.math. uwaterloo.ca /techreports/1999/corrpp-39. ps‖. Row 3 uses affine coordinates and a binary nonadjacent form for the exponent. Row 4 uses mixed Jacobian-affine coordinates and a windowed nonadjacent form for the exponent.

TABLE 4: SAMPLE ELLIPTIC CURVE EXPONENTIATION TIMINGS OVER PRIME FIELDS (IN MILLISECONDS)

Sl. No.	**Processor**	**MHz**	**163 – bit**	**192 – bit**	**256 – bit**	**384– bit**	**521 – bit**
1	Ultra SPARC II	450	6.1	8.7	-	-	-
2	StrongARM	200	22.9	37.7	-	-	-
3	Pentium II	400	-	18.3	42.4	136.4	310.4
4	Pentium II	400	-	2.1	5.1	16.4	27.8

TABLE 5: SAMPLE RSA ENCRYPT/DECRYPT TIMINGS (IN MILLISECONDS).

Sl.No.	**Proce-ssor**	**MHz**	**102-RSA$_d$**	**1024-RSA$_e$**	**2048-RSA$_d$**	**2048-RSA$_e$**
1	Ultra SPARC II	450	32.1	1.7	205.5	6.1
2	Strong ARM	200	188.7	10.8	1273.8	39.1
3	ARM 7TDMI	1	12,070	1180	-	-

In Table 5 RSA$_d$ is the Private-key operation, whereas RSA$_e$ is the Public-key operation. Rows 1 and 2 are from –V. Guptha, S. Guptha and S. Chang, Performance Analysis of Elliptic Curve Cryptography for SSL, ACM Wksp. Wireless Security, Mobicom 2002, Atlanta, GA, Sept. 2002, http://research.sun.com/prolects/crypto/performance.pdf], as above in the elliptic curve timings, Row 3 is from –Performance of RSA on ARM and Palm, http://www.digisec.se/mcrypt_performance. html‖.

8. CONCLUSION

In the past few years ECC has evolved from a fringe activity to a major challenger to the popular RSA. There are many drawbacks in current encryption algorithm in respect of security, real-time performance and so on, and researchers are presenting various algorithms. Among them, the ECC is evolving as an important Cryptosystem, and shows a promise to be an alternative of RSA. Elliptic curves offer major advantages over traditional systems such as increased speed, less memory and smaller key size. Equal security can

be provided by much smaller key length using ECC, to this extent that it can actually be faster than others. In addition, less storage, less power and less memory than other systems make it possible to implement Cryptography in many special platforms such as wireless devices, laptop computers and smart card.

REFERENCES

[1] William Stallings, ―Cryptography and Network Security‖, Prentice Hall, 4th Edition, 2005.

[2] Koblitz, N., A Course in Number Theory and Cryptography‖

[3] V. Guptha, S. Guptha and S. Chang, Performance Analysis of Elliptic Curve Cryptography for SSL, ACM Wksp. Wireless Security, Mobicom 2002, Atlanta, GA, Sept. 2002, http://research.sun.com/prolects/crypto/performance.pdf.

[4] M. Brown et al., Software implementation of the NIST Elliptic curves over prime fields, D. Naccache, Ed., Topics in Cryptology – CT-RSA 2001, LNCS, vol. 2020, Springer – Verlag, 2001, pp. 250 -65.

[5] FIPS Pub 186 – 2, Digital Signature Standard (DSS), Jan.27, 2000. http://csrc.nist.gov/publications/fips/fips186–2/fips186-2-change1 .pdf.

[6] J. Solinas, ‖Generalized Mersenne Numbers,‖ Tech. rep., 1999, http: //www.cacr.math.uwaterloo.ca/techreports/1999/corrpp-39. ps.

[7] Performance of RSA on ARM and Palm,http://www.digisec.se/mcrypt_performance.htm.