A Critical Review on Elliptic Curve Cryptography

Soram Ranbir Singh
Manipur Institute of Technology
Takyelpat, Imphal, India
ranbirsoram@gmail.com

Ajoy Kumar Khan
Assam University
Silchar, India
ajoyiitg@gmail.com

Takhellambam Sonamani Singh
Manipur Institute of Technology
Takyelpat, Imphal, India
sona.cse2013@gmail.com*

***Abstract*: Elliptic curves in the geometric and algebraic parlances have been studied by computer scientists over the last few decades. Elliptic Curve Cryptography or Cryptosystem (ECC) is a public-key cryptosystem. The main advantage and benefit of using ECC instead of using RSA is that it gives equivalent security for a smaller key, thereby consuming less resource and ameliorating performance on the systems. The basics of elliptic curves and their arithmetic are discussed in this paper. We also present experimental results justifying the benefits of using elliptic curve cryptography over RSA in public cryptosystems.**

Keywords: Elliptic Curve, Cryptography, ECDLP, Encryption, Decryption.

I. INTRODUCTION

A mathematical curve is the collection or set of all the points whose coordinates satisfy the polynomial equation of the form $f(x, y) = 0$. The simplest curves are, of course, straight lines and they can be represented by the equation given in (1).

$$ax + by + c = 0 \tag{1}$$

The next simpler possible curves, after straight lines, are conics and they are quadratic forms of the following type as exhibited in (2).

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2}$$

The next possible curves are elliptic curves and they are always cubic. Elliptic curves can be represented by equations of the form given in (3).

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{3}$$

The above equation is known by the name the Weierstrass equation. The variables x, y and the constant a_1, a_2, a_3, a_4, a_6 can be from the domain of integers, real, complex, polynomials, and any other field elements [1].

We take the curve of (3) with the field as K. If we assume that the field K does not have the characteristic of 2, then we can divide the above equation by 2. We then complete the square to get the following (4).

$$\left(y + \frac{a_1x}{2} + \frac{a_3}{2}\right)^2 = x^3 + \left(a_2 + \frac{a_1^2}{4}\right)x^2 + \left(a_4 + \frac{a_1a_3}{4}\right)x + \left(\frac{a_3^2}{4} + a_6\right)$$

$$y_1^2 = x^3 + a_2^1x^2 + a_4^1x + a_6^1 \tag{4}$$

with $y_1 = y + \frac{a_1x}{2} + \frac{a_3}{2}$ and some constants a_2^1, a_4^1, a_6^1. If the characteristic were 2, then 2 is equivalent to 0 in this field K. We will not able to perform the above operation as we cannot divide by zero [1].

Further, if the characteristic is neither 2 nor 3, then we could perform a further substitution by letting $x_1 = x + \frac{a_2^1}{3}$ in (4) as given below.

$$\begin{aligned} y_1^2 &= x^3 + a_2^1x^2 + a_4^1x + a_6^1 \\ &= \left(x_1 - \frac{a_2^1}{3}\right)^3 + a_2^1\left(x_1 - \frac{a_2^1}{3}\right)^2 + a_4^1\left(x_1 - \frac{a_2^1}{3}\right) + a_6^1 \\ &= x_1^3 + \left(a_4^1 - \frac{1}{3}a_2^{1^2}\right)x_1 + \left(\frac{2}{27}a_2^{1^3} - \frac{1}{3}a_2^1a_4^1 + a_6^1\right) \\ &= x_1^3 + Ax_1 + B \end{aligned} \tag{5}$$

for some constants A and B, where $A = a_4^1 - \frac{1}{3}a_2^{1^2}$ and $B = \frac{2}{27}a_2^{1^3} - \frac{1}{3}a_2^1a_4^1 + a_6^1$. We can also transform the curve of (3) by putting straight $X = \frac{x - 3a_1^2 - 12a_2}{36}$ and $Y = \frac{y - 3a_1x}{216} - \frac{a_1^3 + 4a_1a_2 - 12a_3}{24}$ in it and obtain, after simplification, much simpler form as given in (6).

$$Y^2 = X^3 + aX + b \tag{6}$$

II. ELLIPTIC CURVES OVER REAL NUMBERS

Let us consider elliptic curve over real numbers. A simple form of equation is given in (7).

$$y^2 = x^3 + ax + b \tag{7}$$

Let's plot the curve (7) for a=-5, b=1 for x and y in the set of real numbers. We need to find the value of

$$y = \sqrt{x^3 - 5x + 1} \qquad (8)$$

for some values of x. We then plot both the values of y as we have plotted in fig. 1 below.

From this lovely curve, algebra is created to find a method of finding the "addition" of two points on the curve in such a way that the sum we obtained is again another point on the given curve [1]. If we could do it, we can invent an identity element still following the same basic rules of math we used in our high school life.

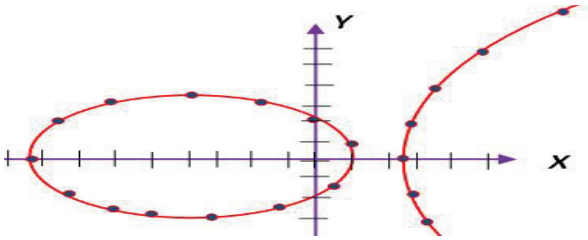

Fig.1. Plot of $y^2 = x^3 - 5x + 1$

The identity element, as represented by **0,** is the point that can be added to any other point on the curve, gives the same point back:

$$P + 0 = P$$

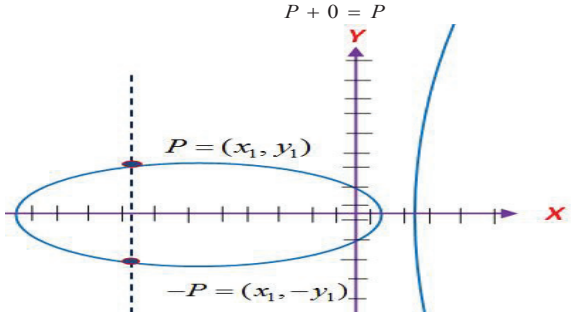

Fig. 2. Arbitrary points P and -P

It is also called "the point at infinity". Some authors also use the symbol ∞ in place of 0 probably to avoid confusion. We seldom use this point in real code [1].

Let's take a point $P = (x, y)$ on any curve. The formula for finding –P is $-P = (x, -y)$.

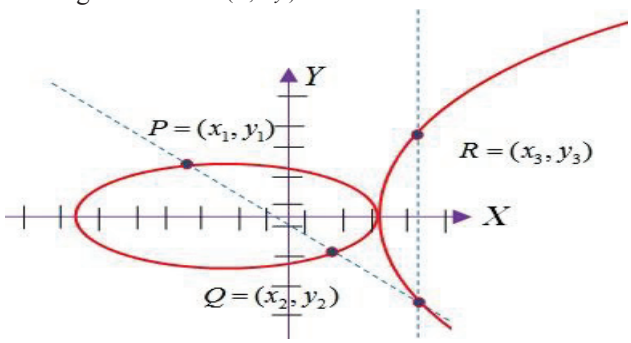

Fig. 3. Adding two points

Now, let's have a look at our fig. 2 and see where the two points, P and –P, lie on the curve. We could see that the two points lie on top of each other. Now, we can add the two points to get the identity element as given in (9).

$$P + (-P) = 0 \qquad (9)$$

If we sketch a line passing through any two points, say, P and −P, the next point on the elliptic curve it passes through is "the point at infinity." In fact, we can add any two given points on a curve by sketching a straight line passing through the two given points and locating the other point at which the line touches or intersects the given curve [1]. Let us take any two points, $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, on an elliptic curve E. Then sum of P and Q, i.e., $P + Q = R(x_3, y_3)$, is defined as follows. First we draw or sketch a straight line passing through the two points, P and Q; this line will touch or intersect the elliptic curve at another point. This point is represented as $-R(x_3, y_3)$.

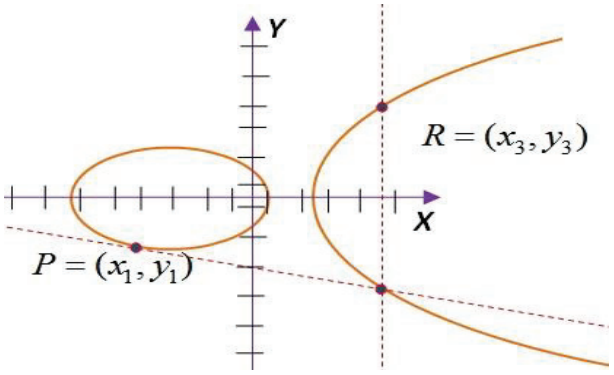

Fig. 4. Doubling a point

We define the negative of the point of intersection as the addition or sum of the two points. That is, $-R(x_3, y_3) = R(x_3, -y_3)$.

The double R, of P is defined as follows. First the tangent line to the curve at P is drawn. This line will surely intersect the curve at another point. Then, $R(x_3, y_3) = 2P(x_1, y_1)$ is the reflection of this point about the x-axis. The multiplication of an elliptic curve point by an integer is converter or transformed into many additions and doubling of many points. Let us take a curve over real number given in (10).

$$y^2 = x^3 - 5x \qquad (10)$$

We take $P = (-1, -2)$ on the curve as given in 10. The multiplication of the point, $P = (-1, -2)$ by an integer, 4 is transformed into additions of many points and all those points are given below.

$$1P + 1P = 2P = \left(\frac{9}{4}, \frac{3}{8}\right), \quad 2P + 1P = 3P = \left(\frac{-121}{169}, \frac{3938}{2197}\right)$$

$$3P + 1P = 4P = \left(\frac{25921}{144}, \frac{-4172959}{1728}\right) = 2P + 2P$$

III. GROUP LAW OF ELLIPTIC CURVES

Let E be a given elliptic curve with the underlying field

as K. There is a rule, known by the name, chord-and-tangent rule for adding any two given points in E(K) to get another point in E(K). Together with the addition operation, the collection of all the points of E(K) forms an abelian group with ∞ (point at infinity) as its identity element [1].

Group law for $y^2 = x^3 + ax + b$

1. Identity element: $P + 0 = 0 + P = P$ for all $P \in E(K)$.
2. Negative of a point: If $P = (x, y) \in E(K)$, then $(x, y) + (x, -y) = 0$. The point $(x, -y)$ is denoted by -*P* and is called the negative of the point, *P*. We must note that -*P* is also a point in $E(K)$. Also, as in normal algebra, $-0 = 0$.
3. Addition of Points: Let $P = (x_1, y_1) \in E(K)$ and $Q = (x_2, y_2) \in E(K)$ where $P \neq \pm Q$ .Then $P + Q = R(x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$.
4. Doubling of a Point: Let $P = (x_1, y_1) \in E(K)$, where $P \neq \pm P$. Then $2P = R(x_3, y_3)$, where $x_3 = \lambda^2 - 2x_1$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{3x_1^2 + a}{2y_1}$

IV. ELLIPTIC CURVRE OVER PRIME GALOIS FIELDS

Real numbers are never used in cryptography as they cannot be manipulated and stored precisely in digital computers. As the number of elements is finite in a Galois field, we can find a unique integer representation for them, which allow us to manipulate the elements in a digital computer.

An elliptic group over the prime Galois Field $E_p(a, b)$ uses an elliptic curve of the form

$$y^2 \pmod p = x^3 + ax + b \pmod p$$

where $a, b \in GF(p), 0 \le x \le p$ and its discriminant should not be zero [1], i.e, $-16(4a^3 + 27b^2) \bmod p \neq 0$.

In mathematics, the discriminant of a function or polynomial is the product of the squares of the differences of the roots. The roots of the polynomial $f(x) = x^3 + ax + b$ are obtained by solving the equation $f(x) = x^3 + ax + b = 0$. Let the roots be r_1, r_2, and r_3. The discriminant will satisfy the relation given below.

$$D = \prod_{i<3}^{3} (r_i - r_j)^2 = (r_1 - r_2)^2 (r_1 - r_3)^2 (r_2 - r_3)^2.$$

For a polynomial is of the form $f(x) = x^3 + ax + b$, the discriminant can be simplified to $-16(4a^3 + 27b^2)$. This discriminant must not be zero for the elliptic curve $y^2 = x^3 + ax + b$ to have three distinct roots.

Fortunately our group law is still applicable if we use finite field elements instead of real numbers. As an example, consider the elliptic curve

$$y^2 = x^3 + 4x + 20 \qquad (11)$$

defined over $GF(29)$. This curve is represented by $E_{29}(4, 20)$.

TABLE 1. Points in $E_{29}(4, 20)$ excluding ∞

(0,7)	(10,25	(3,28)	(15,2)
(1,5)	(19,13)	(5,7	(20,26)
(4,19)	(16,27	(16,2	(4,10)
(20,3)	(5,22	(19,16)	(1,24)
(15,27)	(3,1)	(10,4)	(14,23)
(6,12)	(0,22)	(13,6)	(27,27)
(17,19)	(27,2)	(14,6)	(17,10)
(24,22)	(2,23)	(8,19)	(13,23)
(8,10)	(2,6)	(24,7	(6,17)

Let $P(x_1, y_1) = (1, 5), Q(x_2, y_2) = (20, 3)$. Now, we calculate 2P and R=P+Q as follows:-

$$2P(x_3, y_3) = 1P + 1P = (1, 5) + (1, 5)$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{3X1 + 4}{2X5} = \frac{7}{10} = 7X10^{-1} \pmod{29} = 21$$

$$x_3 = \lambda^2 - 2x_1 = 21^2 - 2X1 = 439 \pmod{29} = 4$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 21(1 - 4) - 5 = -68 \pmod{29} = 19$$

$$\therefore \; 2P(x_3, y_3) = (4, 19)$$

Also,

$$R(x_3, y_3) = P(x_1, y_1) + Q(x_2, y_2) = (1, 5) + (20, 3)$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3 - 5}{20 - 1} = \frac{-2}{19} = -2X19^{-1} \pmod{29}$$

$$= -2X26 \pmod{29} = 6$$

$$x_3 = \lambda^2 - x_1 - x_2 = 6^2 - 1 - 20 = 15 \pmod{29} = 15$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 6(1 - 15) - 5 = -89 \pmod{29}$$

$$= 27$$

$$R(x_3, y_3) = (15, 27)$$

V. ENCRYPTION AND DECRYPTION

ECC can encrypt plaintext message, **M**, into ciphertext, C, and decrypt the ciphertext back into plaintext message, M. The plaintext message **M** is first converted to a single large integer and then mapped to a point on the curve.

A. Key Generation

1. Alice and Bob agree on a domain parameter $D = (q, FR, S, a, b, G, n, h)$ where the generator point is $G = (x_g, y_g)$.
2. Alice chooses an integer n_a as his private key and calculates $P_a = n_a G = (x_a, y_a)$ using the group law.
3. Alice's public key is $P_a = (x_a, y_a)$.
4. Bob also chooses an integer n_b as his private key and calculates $P_b = n_b G = (x_b, y_b)$ according to group law.
5. Bob's public key is $P_b = (x_b, y_b)$.

B. Encryption

Alice is dispatching the message $P_m = (x_m, y_m)$ to Bob. His algorithm is given below.

1. Alice picks up a random number k.
2. He evaluates the two points, $c_1 = kG$ and $c_2 = P_m + kP_b$.
3. Alice sends the pair of points, $P_c = \{c_1, c_2\}$ as cipher to Bob.

C. Decryption

Bob receives the ciphertext, $P_c = \{c_1, c_2\}$ from Alice. Bob reconstructs the original message as follows:

He multiplies c_1 by his private key n_b and subtracts it from c_2. That is, he calculates

$$c_2 - n_b c_1 = (P_m + kP_b) - n_b(kG) = (P_m + kn_b G) - n_b kG = P_m = (x_m, y_m)$$

D. ECC Encryption and Decryption Example

Alice chooses an elliptic curve of the form $y^2 = x^3 + ax + b \bmod p$ where

a=1157920892103562487626974469494075735300861434152903141955336313088670978539 48,

b=4105836372515214212932612978004726840911444101599372555483525631403946740129 1,

p=1157920892103562487626974469494075735300861434152903141955336313088670978539 51. The base point or generator is $G = (x_g, y_g)$, where

x_g=4843956129390645175905258525279791420276294952604174799584408071708240463528 6 and

y_g=3613425095674979579858512791958788195661110667298501507187719825356841440510 9. The order of the base point or the generator is n such that $nG = \infty$. Here the value of

n=1157920892103562487626974469494075735299969552241357603424222590610685120443 69.

Assume Alice wants to send the message "**IEEE Conferences in India**" to Bob. He first converts it to an integer value which is given as

4599269365104815827945572879896355396563704505503943871962 57. Next he encodes it to a point (normally x-coordinate) on the curve. The encoded point is $P_m(x_m,y_m)$ where

x_m=**13797808095314447483836718639689066189691113516511831615887710**,

y_m=6342517782103783068568442641430160810765252773757222017927295832000107817787 7. He uses Bob's public key to encrypt it. Assume the private key of Bob is

n_b=5699869819141673937078284284221734741505388.

The public key as announced by Bob is $P_b(x_b, y_b) = n_b G(x_g, y_g)$ where

x_b=5560588716881227967077118286652631257008617096275013591922030982670690104421 3 and

y_b=8758654858369616145062982884568330793005691659922661445833686364114898284327 0.

Alice chooses a random integer k=2345 and computes the ciphertext pair of points P_c using Bob's public key P_b:

$P_c = \{C_1, C_2\} = [(kG), (P_m + kP_b)]$

The ciphertext calculated by Alice is two pair of points, $P_c = \{C_1, C_2\}$ where $C_1=(x_1,y_1)$, $C_2=(x_2,y_2)$ and

$\mathbf{x_1}$=5426006802596267112236437744297025462012174190897515068886426183949850916 4265,

$\mathbf{y_1}$=8037541005771019073464399336982769605968765237194462512812205495394307 1012232,

$\mathbf{x_2}$=9344950228499248855555991901178323098396358957408269483110842289431685 7328680,

$\mathbf{y_2}$=8000525311299448272506871001170514379421644378551223769640475739481453 8545998.

Bob recover the plaintext from the ciphertext using

$$P_m(x_m, y_m) = c_2 - n_b c_1 = (P_m + kP_b) - n_b(kG) = (P_m + kn_b G) - n_b kG$$

where

x_m=**13797808095314447483836718639689066189691113516511831615887710**,

y_m=6342517782103783068568442641430160810765252773757222017927295832000107817787 7. The x-coordinate point is transformed to the integer value

4599269365104815827945572879896355396563704505503943871962 57, where it is mapped to the text message, "**IEEE Conferences in India**".

One of the complicated considerations in elliptic curve cryptography is the transforming of arbitrary plaintext

message into a point on the elliptic curve. Here we transform the plaintext message as x_i, the x-coordinate of a point, $P_m = (x_i, y_i)$, on an elliptic curve E of the form $y^2 = x^3 + ax + b$ over $GF(p)$.

VI. The Performance of Elliptic Curve Cryptography over RSA

In this section we compare public key generation times, the encryption and decryption implementation speeds, the ciphertext sizes and digital signature signing and verification times for key sizes that are said to be equal in terms of security. The plaintext message that we used in the encryption and decryption algorithm is (="IEEE Conferences in India") of 25 bytes. The ECC Private Key and the RSA public exponent we used in all the experiments are 569986981914167393707828428422173474150538 8 (randomly chosen) and 65537(technically chosen) respectively. All the algorithms were implemented in Java 7 on our Intel 3.10 GHz and 4GB of RAM. In table 2 below, we give the encryption and decryption operating time takens in nanosecond. The result is the average of four test runs.

TABLE 2: Encryption and Decryption Time Taken

Security level(bits)	Algorithms	Encryption Time (ns)	Decryption Time (ns)
112	ECC-224	1450239	10231465
	RSA-2048	2840752	260099529
128	ECC-256	1848262	11607045
	RSA-3072	4405311	593957338
192	ECC-384	2061037	12794513
	RSA-7680	17218556	**8837759875**

The encryption time takens of ECC algorithms given in table 2 do not consider time required to map the message to an elliptic curve point. The time taken in nanosecond to map the message (=**IEEE Conferences in India**) to an elliptic curve point is tabulated in table 3 below. The result is the average of ten test runs.

TABLE 3: Message to Point Mapping Time Taken

Security level(bits)	Algorithms	Message to Point Mapping Time (ns)
112	ECC-224	4541096
128	ECC-256	6128053
192	ECC-384	26145365

In table 4 below, we give the public key generation time taken in nanosecond. The public key generation time taken of RSA includes the primality testing part as well. Each result is the average of five test runs.

TABLE 4: Public Key Generation Time Taken

Security level(bits)	Algorithms	Public Key Generation Time (ns)
112	ECC-224	30239910
	RSA-2048	2825151941
128	ECC-256	32403603
	RSA-3072	10270890931
192	ECC-384	32457264
	RSA-7680	**102067678287**

TABLE 5: Ciphertext Size in Byte

Security level(bits)	Algorithms	Ciphertext Size(byte)
112	ECC-224	248
	RSA-2048	616
128	ECC-256	248
	RSA-3072	925
192	ECC-384	256
	RSA-7680	2311

In table 5 above, we give the ciphertext sizes in bytes of ECC and RSA algorithms. For ECC algorithms, the ciphertext sizes mean two pair of points. The result is the average of five test runs.

In table 6 below, we give the time taken to perform ECC signature signing and verification operations with that of RSA's. The message signature we used for signing in both the algorithms is "**IEEE Conferences in India**".

VII. Elliptic Curve Discrete Logarithm Probelm

Let E be an elliptic curve defined over a finite field, K and let, P be a point on E and k is an integer. The point $Q = kP$ can be computed from P very easily by repeated point additions of P. However, it is very hard to determine the value of k knowing the two points: kP and P and this gives the Elliptic Curve Logarithm Problem (ECDLP) which is stated as "Given a point P and the point $Q = kP$, on the curve E, find the value of the integer k". This integer k is

called the elliptic curve discrete logarithm [1] of Q to the base P, denoted as $k = \log_P Q$.

TABLE 6: Signature Signing and Verification Timings

Algorithms	Signature Generate Time (ns)	Signature Verify Time (ns)
ECC-224	39564742	44990073
RSA-2048	74699908	1703145
ECC-256	46642822	54761921
RSA-3072	185208775	3382025
ECC-384	86548193	110636252
RSA-7680	**2293216586**	18538609

To demonstrate the ECDLP, we take an elliptic curve, *E*, given by $y^2 = x^3 + 4x + 20$ over F_{29}. We are given the relation $Q(20,3) = kP(1,5)$. Let us find the discrete logarithm *k* of $Q = (20,3)$ to the base $P = (1,5)$. A raw method to find the integer k is to compute multiples of P until one of them is equal to Q.

We have, $P(x_1, y_1) = (1,5)$. Now, we calculate 2P as follows:-

$$2P(x_3, y_3) = 1P + 1P = (1,5) + (1,5)$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} = \frac{3X1 + 4}{2X5} = \frac{7}{10} = 7X10^{-1} (\mathrm{mod}\ 29)$$

$$= 7X3 = 21$$

$$x_3 = \lambda^2 - 2x_1 = 21^2 - 2X1 = 439 (\mathrm{mod}\ 29) = 4$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 21(1-4) - 5 = -68 (\mathrm{mod}\ 29)$$

$$= 19$$

$$\therefore\ 2P(x_3, y_3) = (4,19)$$

As $2P(x_3, y_3) = (4,19) \neq Q(20,3)$, we again calculate 3P.

$$3P(x_3, y_3) = 1P(x_1, y_1) + 2P(x_2, y_2)$$

$$= (1,5) + (4,19)$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{19-5}{4-1} = \frac{14}{3} = 14X3^{-1} (\mathrm{mod}\ 29)$$

$$= 14X10 (\mathrm{mod}\ 29) = 24$$

$$x_3 = \lambda^2 - x_1 - x_2 = 24^2 - 1 - 4 = 571 (\mathrm{mod}\ 29) = 20$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = 24(1-20) - 5$$

$$= -461 (\mathrm{mod}\ 29) = 3$$

$$3P(x_3, y_3) = (20,3)$$

Since, $3P(x_3, y_3) = (20,3) = Q(20,3)$, the discrete logarithm of Q to the base P is $k = 3$. In other words, $\log_P Q = \log_{(1,5)} (20,3) = 3$.

CONCLUSION

Elliptic Curve Cryptosystems are based on some Galois fields. We discuss a broad view of elliptic curves, the group laws of elliptic curves and elliptic curve discrete logarithm problem. We also gave the performance of the computer implementation results of ECC and RSA. The encryption process in RSA is favorable even for very large key sizes. However, decryptions process in RSA is CPU-intensive and very much time consuming. Both the encryption and decryption speeds of the ECC algorithms are optimal even for very large key sizes. The RSA signature signing is slower than verification whereas ECC signature signings are generally faster than the verifications. The public key generation time taken of RSA algorithm is very greater than that of ECC's. These implementation results appeal us to use elliptic curve cryptography as a replacement for RSA.

REFERENCES

[1] Silverman, Elliptic Curve, Springer International Edition, 2010.
[2] Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography, 2nd Edition, 2008.
[3] Blake,Seroussi, and Smart, Elliptic Curves in cryptography, Cambridge University Press,1999.
[4] Michael Rosing, Implementing Elliptic Curve Cryptography, Manning Publisher.
[5] Elisabeth Oswald, Introduction to Elliptic Curve Cryptography, 2005.
[6] Joseph H. Silverman and John T Tate, Rational Points on Elliptic Curves, Springer.
[7] William Stallings, Cryptography & Network Security, PHI, 2006.
[8] Don B. Johnson, Alfred J. Menezes, Elliptic Curve DSA (ECDSA): An Enhanced DSA.
[9] Jean-Yves Chouinard, Notes on Elliptic Curve Cryptography, 2002.
[10] Ranbr Soram and Engudam Sanahal Meitei, International Symposium on Advanced Computing and Communication (ISACC), 2015.
[11] Kristin Lauter, The Advantages of Elliptic Curve Cryptography for Wireless Security, Microsoft Corporation.
[12] G. Vennila, M. Manikandan and S. Aswathi, Performance analysis of point multiplication algorithms in ECDH for an end-to-end VoIP network, INDICON, 2015.
[13] P. Mathew , P. Jilna , P. P. Deepthi, Efficient implementation of EC based key management scheme on FPGA for WSN, International Conference on Telecommunication Systems Services and Applications (TSSA), 2015.