Some Algorithms of Various Projective Coordinate Systems for ECC Using Ancient Indian Vedic Mathematics Sutras

Manoj Kumar, Ankur Kumar

Abstract— In this present approach, Some Algorithms of Various Projective Coordinate Systems for ECC (Elliptic Curve Cryptography) using AIVM (Ancient Indian Vedic Mathematics) sutras, has been studied. This work explained some useful Vedic sutra for multiplication calculation in cryptographic operations. In this paper, we have used some Vedic Mathematics Sutra to get minimum steps in the calculation of the addition algorithm, doubling algorithm and for improving the speed of processing time in the cryptographic operations, such as point addition, point doubling which occurs in the Elliptic curve cryptography over projective coordinate systems (Standard Projective, Jacobian Projective, Lopez-Dahab Projective). The coding and synthesis are done in MATLAB. The results proved that the Vedic Mathematics based schemes show better performance compared to the conventional method. The total delay in computation is reduced by Vedic mathematics Sutras (Urdhva-Tiryagbhyam, Dvandva-Yoga) with the help of MATLAB software.

Index Terms— Dvandva-Yoga, Elliptic Curve Cryptography, Jacobian Projective, Lopez-Dahab Projective, Point addition, Point doubling, Standard Projective, Urdhva-Tiryagbhyam.

1 INTRODUCTION

lliptic curve cryptography (ECC) is an approach which is based on public key cryptography. Elliptic curves (EC) were proposed first time by Koblitz in the terms of Cryptography in 1987 [9]. In these days all countries are totally dependent on the internet, for example, secure telephony, electronic mail, mobile internet, electronic commerce, communication source and most important is online banking transaction, and many more. So security plays the most important role in the system of data transfer or any information over the whole network. ECC systems give information security like data integrity, confidentiality, and authentication of data and soon. ECC is all about the study of techniques for sending information or messages secretly that messages read by one and only recipient. A Sender can encrypt the information and receiver can decrypt that information after transmission to get real information. RSA and ECC are working safely in standard algorithm, which is totally depends on public key cryptography (PKC) for good security in networks. RSA algorithm has a large key size but ECC has a small key size so security provider prefers ECC over RSA [8, 16, 22,]. AIVM sutras can be used to reduce the time of calculation involving squaring and multiplication operation which result speed up ECC based cryptosystems. The performance of most ECC system is overall decided by a proficient implementation of operations arithmetically. AIVM sutras provide proficient techniques for all computations of ECC [2, 14, 16, 18].

- *Manoj Kumar is currently an assistant professor in the department of Mathematics and statistics at the Gurukula Kangri Vishwavidyalaya Haridwar India, PH +91 8755386009. E-mail: sdmkg1@gmail.com*
- *Ankur Kumar is currently pursuing Ph.D. in the department of Mathematics and statistics at the Gurukula Kangri Vishwavidyalaya Haridwar India, PH-+91 9997760427. E-mail: ankurgkv99@gmail.com*

2 RELATED WORK

In this section, we will discuss the related work done by researchers on cryptographic applications using AIVM (Ancient Indian Vedic Mathematics). In 2014, also by Thomas et al [21] determined in his paper, on the different algorithm for ECC that, the ECC with Karatsuba Multiplier gives high speed, minimum area (save 51 %), minimum time (save 46 %) and reduce the number of the clock cycle in the hardware implementation of algorithms. Gaikwad and Chavan [5] in 2015, considered in his review paper that the AIVM sutras can be used for fast signal processing and results give an improvement in speed reduction in consumption of power, area, and complexity, etc. Salim and Mandaogade [15] in 2015, intended that implementation of RSA (Rivest, Shamir, Adleman) cryptography using AIVM sutras. They explained in his review paper about the time variation and cost variation in cryptographic operations to maintaining the integrity of data or information. In 2016, Warang and Tambe [23] proposed that high-speed complex multiplier with AIVM sutras is an effective tool and AIVM is so effective a tool it helps to reduce so much time and increase computational speed. In this approach, they have explained the use of Urdhva-Tiryagbhyam sutra for multiplications in cryptographic operations. In 2016, Pinagle [13] well described cryptography and ECC with defining the model problem for prime curves and implementation of the result by 'C' programming. A review of AIVM sutras by Shembalkar [17] in 2017, they explained sixteen sutras of AIVM such as Urdhva-Tiryagbhyam, Nikhilam, and Dvandva-Yoga so on and the result is seen to be faster processing in speed and fewer areas in circuits. Deepa and Marimuthu [3] in 2018 they explained squaring by AIVM sutras for digital signals and cryptographic operations and results show excellent performance in processing and reduce the delay. An analysis of AIVM based cryptography algorithms by Lisha and Monoth [10] in 2018, they proposed about performance analysis of AIVM sutras based algorithms and results show these algorithms save

processing time and hardware resources.

3 ELLIPTIC CURVE CRYPTOGRAPHY

In this section we will through some light on basic nomenclature arithmetic background of ECC. Following symbols are used to define elliptic curves [9]

- $GF(p)$ is Galois Field over prime p.
- p is prime number greater than 3.
- a, b is fixed real number.
- (x, y) is point on the elliptic curve E

Using above terminology, the elliptic curve E defined [9] as the set order pair (x, y) on the curve $y^2 = x^3 + ax + b \pmod p$ Pleasing the discernment equation $4a^3 + 27b^3 \neq 0 \pmod p$. Set theoretically an elliptic curve can be represented as:

$$E = \{(x, y) : y^2 = x^3 + ax + b \pmod p \quad and \quad 4a^3 + 27b^3 \neq 0 \pmod p\}.$$

The elliptic curve $E(F_p)$ explained as [9]:

- The additive identity property explained as: $P + O = O + P$ where P belongs to $E(F_p)$.
- The additive inverse property explained as: $(x, y) + (x, -y) = O$ where (x, y) belongs to $E(F_p)$.
- The sum of two distinct points $P + Q = R$ are shown in figure 1; where $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ is R (x_3, y_3) and x_3, y_3, λ are

$$\begin{bmatrix} (\lambda^2 - x_1 - x_2), \\ (\lambda(x_1 - x_3) - y_1), \\ (y_2 - y_1)/(x_2 - x_1) \end{bmatrix}$$ respectively.

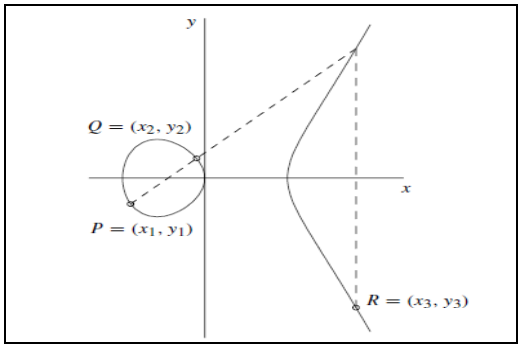

Figure 1 : Addition (P+Q=R) [9]

Thre point doubling of a point $2P = R$ are shown in figure 2, where x_3, y_3, λ are

$\{(\lambda^2 - 2x_1), (\lambda(x_1 - x_3) - y_1), (3x_1^2 + a)/2y_1\}$ Respectively.

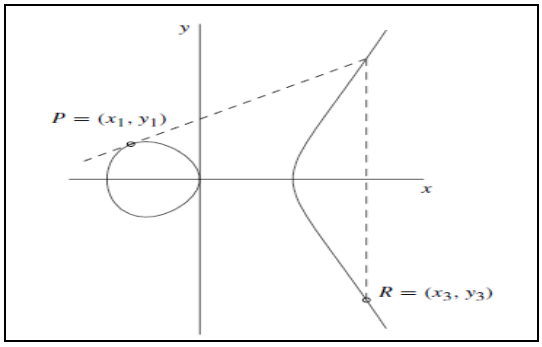

Figure 2: Doubling (R = 2P) [9]

4 ANCIENT INDIAN VEDIC MATHEMATICS

Across India, the graph of the research is increasing speedily in the field of AIVM (Ancient Indian Vedic Mathematics) which contain the sixteen sutras and fourteen sub sutras or formulae. These useful Sutras have been conventionally used for square, cube, divide and multiplications in a decimal number and also in binary number. In this paper, we applied some new ideas to make the proposed algorithm. These algorithms can work with the decimal and binary number system. In this section, we will discuss some useful techniques and methods to improve the performance of ECC based on Projective Coordinate System. The first technique is Urdhva-Tiryagbhyam and second is Dvandva-Yoga of AIVM [20, 22].

4.1 Urdhva-Tiryagbhyam Sutra of AIVM

The Sutras of Ancient Indian Vedic Mathematics (AIVM) is using enormously in the research area of Mathematics. For multiplication we have used AIVM sutra name is Urdhva-Tiryagbhyam which gives the best output in the operations of ECC.

This Sutra reduces bits and steps in multiplication and saves much time in encryption and decryption of ECC. Compression between AIVM based multiplier, name is Urdhva-Tiryagbhyam and other multipliers, name is Booth, Array and many more, we observed that AIVM multiplier has not much delay and save time and power in the ECC operations, Naturally by the use of standard or classical method in multiplication we can get results but number of operation in classical method is too high almost operations for m bit digit integers so it looks complicated.

Another multiplication method is the Karatsuba method it requires the number of operation for two integers of m-bit. So this is a complicated method of multiplication and other is Karatsuba technique (method) gives slow output for a small input in any operations comparatively classical technique (methods) of multiplication due to the repetition of overhanging operations. To manipulate this type of issue best sutra of AIVM Urdhva-Tiryagbhyam technique for multiplications can be applied and get the best result easily [22].

Multiplication of three digits Numbers

$$(p_1x_1^2 + q_1x_1 + r_1)(s_1x_1^2 + t_1x_1 + u_1)$$

$$= p_1s_1x_1^4 + (p_1e_1 + q_1s_1)x^3$$

$$+ (p_1u_1 + q_1t_1 + r_1s_1)x_1^2 + (q_1u_1 + t_1r_1)x_1 + r_1u_1$$

Where the base of the number system is x_1 .

4.2 Dvandva-Yoga Sutra of AIVM

For squaring by the Dvandva-Yoga, any binary or decimal number, a purposeful architectonics can rise up its performance and best output than other architecture's multiplier. Using Dvandva-Yoga (D_Y) algorithm and rule for squaring of binary or decimal numbers from the AIVM sutras is explained as [20, 22]:

- To calculate Dvandva-Yoga (D_Y) of a number which contains single digit Dvandva-Yoga expressed that it is the square of that number, Dvandva-Yoga of p_1 is p^2 .
- To calculate Dvandva-Yoga (D_Y) of a number which contains two digits, Dvandva Yoga expressed that, it's double the multiplication of both digits of that number, Dvandva-Yoga of $p_1 q_1$ is $2 * p_1 * q_1$.
- To calculate Dvandva-Yoga (D_Y) of numbers which contain three digits, Dvandva-Yoga expressed that, it's got double the product of first and third number and gives the square of that number which is placed in the middle, Dvandva-Yoga of $p_1 q_1 r_1$ is $2 * p_1 * r_1 + q_1^2$.

5 MATHEMATICAL BACKGROUND OF PROJECTIVE COORDINATE SYSTEM

The elliptic curve contains a pair of coordinate systems first is affine and other is projective. For inversion, affine coordinate system has needed the inversion at the time of doubling and the addition of the points, which are so expensive in terms of time, area and speed. We have used coordinate system to remove very demanding operating inversion operation. Elliptic curve operations like addition, doubling needs a fixed number of Square, addition, modular multiplications, shifts, and primary needed fundamental operations. All these kind of operations like, multiplications and squaring depends on the presentation of an elliptic curve for manage the running time in ECC operations [1, 6].

The affine plane $A_F^2 = \{(x, y) \in F \times F\}$ for coordinates (x, y) are mapped to the projective plan $P_F^3 = (X, Y, Z)$ these coordinate s belongs to $F \times F \times F$. The Coordinate s (x, y) of the affine plane $A_F^2 = \{(x, y) \in F \times F\}$ are mapped to the Coordinates (X, Y, Z) of projective plane $P_F^3 = \{(X, Y, Z) \in F \times F \times F\}$ with this rule:

$$(X, Y, Z) = (x.Z^c, y.Z^d, 1) \; or$$

$$X = xZ^c \; and \; Y = yZ^d \qquad \ldots (1)$$

5.1 Point Addition in Projective Plane:

Case I: If $x_1 \neq x_2$ then we have

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{Y_2 - \frac{Y_1}{Z_1^d}}{X_2 - \frac{X_1}{Z_1^c}}$$

It is obvious from the above expression that λ exist because $x_1 \neq x_2$.

Now the point P_3 can be calculated as:

$$x_3 = \lambda^2 - x_1 - x_2 = \left(\frac{Y_2 - \frac{Y_1}{Z_1^d}}{X_2 - \frac{X_1}{Z_1^c}} \right)^2 - \frac{X_1}{Z_1^c} - X_2, \quad y_3 = \lambda(x_1 - x_3) - y_1$$

$$= \left(\frac{Y_2 - \frac{Y_1}{Z_1^d}}{X_2 - \frac{X_1}{Z_1^c}} \right) \left(\frac{X_1}{Z_1^c} - x_3 \right) - \frac{Y_1}{Z_1^d}.$$

5.2 Points Doubling in Projective Plane:

Case II: If $x_1 = x_2$ then we have for point doubling we can take $P_1 = P_2$ then

$P_3 = P_1 + P_2 = 2P_1 = (X_3, Y_3, Z_3)$. We have

$$\lambda = \frac{3x_1^2 + a}{2y_1} = \left(\frac{3\left(\frac{X_1}{Z_1^c}\right)^2 + a}{2\frac{Y_1}{Z_1^d}} \right), \quad x_3 = \left(\frac{3\left(\frac{X_1}{Z_1^c}\right)^2 + a}{2\frac{Y_1}{Z_1^d}} \right)^2$$

$$-2\frac{X_1}{Z_1^2} = \frac{(3X_1^2 + aZ_1^{2c})^2 Z_1^{2d} - 8Z_1^{3c} X_1 Y_1^2}{4Z_1^{4c} Y_1^2}$$

Evidently λ exists if $y_1 \neq 0$, so we get

$$x_3 = \lambda^2 - 2x_1 = \frac{(3X_1^2 + aZ_1^{2c})^2 Z_1^{2d}}{4Z_1^{4c} Y_1^2} - 2\frac{X_1}{Z_1^c}$$

$$= \frac{(3X_1^2 + aZ_1^{2c})^2 Z_1^{2d} - 8Z_1^{3c} X_1 Y_1^2}{4Z_1^{4c} Y_1^2},$$

$$y_3 = \lambda(x_1 - x_3) - y_1 = \lambda(3x_1 - \lambda^2) - y_1$$

$$= \frac{(3X_1^2 + aZ_1^{2c})^2 Z_1^d}{2Z_1^{2c} Y_1} \left(3\frac{X_1}{Z_1^c} - \frac{(3X_1^2 + aZ_1^{2c})^2 Z_1^{2d}}{4Z_1^{4c} Y_1^2} \right) - \frac{Y_1}{Z_1^d}$$

$$= \frac{12X_1 Y_1^2 (3X_1^2 + aZ_1^{2c})^2 Z_1^{3c+2d} - (3X_1^2 + aZ_1^{2c})^3 Z_1^{4d} - 8Z_1^{6c} Y_1^4}{8Z_1^{6c+d} Y_1^3}$$

6 Algorithms of Standard, Jacobian Lopez Dahab Projective Coordinate System over the Curve $y^2 = x^3 + ax + b$

6.1 Standard Projective Coordinates (SPC) System over the curve $y^2 = x^3 + ax + b$

This SPC system is used to show the EC points over the curves $y^2 = x^3 + ax + b$. SPC system gives a better speed over affine Coordinate s system when field inversions' value is automatically higher multiplication of field. In the SPC system the triple (X, Y, Z) express the affine point (X / Z, Y / Z).

The SPC system with $c = 1$ and $d = 1$ in equation (1), the elliptic curve E can be reworded as $E : Y^2Z = X^3 + aXZ^2 + bZ^3$.

Case I. SPC system with $c = 1$ and $d = 1$, the addition of two points given by

$P_1 + P_2 = P_3 = (X_3, Y_3, Z_3)$

Where

$$X_3 = \begin{bmatrix} (X_1Z_2 - X_2Z_1)(Y_2Z_1 - Y_1Z_2)^2 Z_1Z_2 \\ -(X_1Z_2 + X_2Z_1)(X_1Z_2 - X_2Z_1)^3 \end{bmatrix}$$

$$Y_3 = \begin{bmatrix} (Y_2Z_1 - Y_1Z_2)(X_2Z_2 - X_2Z_1)^2 X_1Z_2 - (Y_2Z_1 - Y_1Z_2)^3 Z_1Z_2 \\ +(X_1Z_2 + X_2Z_1)(X_1Z_2 - X_2Z_1)^2 (Y_2Z_1 - Y_1Z_2) \\ -(X_1Z_2 - X_2Z_1)^3 Y_1Z_2 \end{bmatrix}$$

$$Z_3 = \left[(X_1Z_2 - X_2Z_1)^3 Z_1Z_2 \right].$$

6.1.1 Algorithm:
Addition of two points in Standard Projective Coordinate (SPC) system:

Input: $P_1 = (X_1, Y_1, Z_1)$ *and* $P_2 = (X_2, Y_2, Z_2)$

Output: $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$

1. $A = X_1 \cdot Z_2$
2. $B = X_2 \cdot Z_1$
3. $D = Y_1 \cdot Z_2$
4. $E = Y_2 \cdot Z_1$
5. $J = Z_1 \cdot Z_2$
6. $K = Y_1 \cdot Z_2$
7. $L = X_1 \cdot Z_2$
8. $C = A - B$
9. $F = E - D$
10. $G = A + B$
11. $H = F^2 \cdot J - G \cdot C^2$
12. $I = C \cdot L - H$
13. $X_3 = C \cdot H$
14. $Y_3 = F \cdot I - C^3 \cdot K$
15. $Z_3 = J \cdot C^3$
16. $Return(X_3 : Y_3 : Z_3)$.

Finally, we can calculate $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$ where $X_3 = C \cdot H$, $Y_3 = F \cdot I - K \cdot C^3$ and $Z_3 = J \cdot C^3$.

Here we can use Urdhva-Tiryagbhyam and Dvandva-Yoga technique to evaluate

$C \cdot H$, $F \cdot I, K \cdot C^3$ *and* $J \cdot C^3$.

Case II. SPC system with $c = 1$ and $d = 1$, the doubling of the point P_1 is given by

$P_3 = 2P_1 = (X_3, Y_3, Z_3)$ Where

$X_3 = 18X_1^4 + 2a^2Z_1^4 + 12aX_1^2Z_1^2 - 8X_1Y_1^3Z_1^2$

$Y_3 = 3X_1^2Y_2Z_1^3 + Y_2Z_1^5 - 8Y_1^4Z_1^2$ *and* $Z_3 = 8Y_1^3Z_1^3$

6.1.2 Algorithm:
Doubling of a point in Standard Projective Coordinate (SPC) system:

Input: $P_1 = (X_1, Y_1, Z_1)$, a

Output: $P_3 = (X_3, Y_3, Z_3) = 2P_1$

1. $A = 3X_1^2 + aZ_1$
2. $B = Y_1 \cdot Z_1$
3. $F = Y_1 \cdot X_1$
4. $G = Y_1 \cdot B$
5. $C = F \cdot B$
6. $D = A^2 - 8C$
7. $E = 4C - D$
8. $X_3 = 2B \cdot D$
9. $Y_3 = A \cdot E - 8G^2$
10. $Z_3 = (2B)^3$
11. $Return(X_3 : Y_3 : Z_3)$.

Here $B \cdot D$, $A \cdot E$, B^3, *and* G can be calculated using Urdhva-Tiryagbhyam and Dvandva-Yoga technique.

6.2 Jacobian Projective Coordinate (JPC) System over The curve $y^2 = x^3 + ax + b$

This JPC system is used to show the EC points over the curves $y^2 = x^3 + ax + b$. JPC system gives a better speed over affine Coordinate s system when field inversions' value is automatically higher multiplication of field. In the JPC system the triple (X, Y, Z) express the affine point $\left(X / Z^2, Y / Z^3\right)$. The JPC system with $c = 2$ and $d = 3$ in equation (1), the elliptic curve E can be reworded as $E : Y^2 = X^3 + aXZ^4 + bZ^6$.

Case I. JPC system with $c = 2$ and $d = 3$, the addition of two points given by $P_1 + P_2 = P_3 = (X_3, Y_3, Z_3)$ where

$$X_3 = (Y_2 Z_1^3 - Y_1)^2 - (X_1 + X_2 Z_1^2)(X_2 Z_1^2 - X_1)^2$$

$$Y_3 = (Y_2 Z_1^3 - Y_1)\left(2X_1 + X_2 Z_1^2\right)\left(X_2 Z_1^2 - X_1\right)^2$$

$$- (Y_2 Z_1^3 - Y_1)^3 - Y_1\left(X_2 Z_1^2 - X_1\right)^3$$

$$Z_3 = (X_2 Z_1^2 - X_1) Z_1.$$

6.2.1 Algorithm: Addition of points in Jacobian Projective Coordinate (JPC) System:

1. $A = Z_1^2$

Input: $P_1 = (X_1, Y_1, Z_1), P_2 = (X_2, Y_2, Z_2)$

1. $A = Z_1^2$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

3. $C = X_1 \cdot A$
2. $B = Z_1^2 \cdot A$

4. $D = Y_2 \cdot B$
5. $E = C - X_1$

6. $F = D - Y_1$
7. $G = E^2$

8. $H = G \cdot E$
9. $I = X_1 \cdot G$

10. $J = I - X_3$
11. $X_3 = F^2 - H - 2I$

12. $Y_3 = F \cdot J - Y_1 \cdot H$
13. $Z_3 = Z_1 \cdot E$

14. $Return(X_3 : Y_3 : Z_3)$.

Finally we can calculate,

$P_3 = P_1 + P_2 = (X_3, Y_3, Z_3), where$

$X_3 = F^2 - H - 2I$, $Y_3 = F \cdot J - Y_1 \cdot H$,

$and\ Z_3 = Z_1 \cdot E$

Here we can use Urdhva-Tiryagbhyam technique in all multiplications and Dvandva-Yoga technique for all squares.

Case II. JPC system with $c = 2$ and $d = 3$, the point doubling of a point P_1 is given by $P_3 = 2P_1 = (X_3, Y_3, Z_3)$, where

$$X_3 = (3X_1^2 + aZ_1^4)^2 - 8X_1 Y_1^2$$

$$Y_3 = 12 X_1 Y_1^2 (3X_1^2 + aZ_1^4) - (3X_1^2 + aZ_1^4)^3$$

$$- 8Y_1^4 \ and \ \ Z_3 = 2Y_1 Z_1$$

6.2.2 Algorithm: Doubling of a point in Jacobian Projective Coordinate (JPC) System:

1. $A = Y_1^2$

2. $B = 4X_1 \cdot A$
3. $C = 8A^2$

4. $D = 3X_1^2 + a.Z_1^4$
5. $X_3 = D^2 - 2B$

6. $Y_3 = D \cdot (B - X_3) - C$
7. $Z_3 = 2Y_1 \cdot Z_1$

8. $Return(X_3 : Y_3 : Z_3)$.

Finally, we can calculate $P_3 = (X_3, Y_3, Z_3) = 2P_1$ where $X_3 = D^2 - 2B$, $Y_3 = D \cdot (B - X_3) - C$, $Z_3 = 2Y_1 \cdot Z_1$ and finally. Here $X_1^2, X_1.Y_1^2, Y_1^4, A^2 \ and \ Y_1 \cdot Z_1$ can be calculated by Urdhva-Tiryagbhyam and Dvandva-Yoga technique.

6.3 Lopez-Dahab Projective Coordinate (LDPC) System over the curve $y^2 = x^3 + ax + b$

This LDPC system is used to show the EC points over the curves $y^2 = x^3 + ax + b$. LDPC system gives a better speed over affine Coordinate s system when field inversions' value is automatically higher multiplication of field. In LDPC system the triple (X, Y, Z) express the affine point $(X / Z, Y / Z^2)$ The LDPC system with $c = 1$ and $d = 2$ in equation (1), the elliptic curve E can be reworded as $E : Y^2 = XZ + aXZ^3 + bZ^4$.

Case I. LDPC system with $c = 1$ and $d = 2$, the addition of two points given by $P_1 + P_2 = P_3 = (X_3, Y_3, Z_3)$ where

$$X_3 = (Y_2 Z_1^3 - Y_1)^2 - Z_1 (X_1 + X_2 Z_1)(X_2 Z_1^2 - X_1)^2$$

$$Y_3 = Z_1^2 (Y_2 Z_1^2 - Y_1)(2X_1 + X_2 Z_1)\left(X_2 Z_1^2 - X_1\right)^3$$

$$- Z_1 \left(X_2 Z_1 - X_1\right)(Y_2 Z_1^2 - Y_1)^3 - Y_1 Z_1^2 \left(X_2 Z_1^2 - X_1\right)^4$$

$$Z_3 = Z_1^2 (X_2 Z_1 - X_1)^2$$

6.3.1 Algorithm:

Addition of points in Lopez-Dahab Projective Coordinate (LDPC) System:

Input: $P_1 = (X_1, Y_1, Z_1)$ *and* $P_2 = (X_2, Y_2, Z_2)$

Output: $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$

1. $A = Z_1^2$

2. $B = A \cdot Y_2$
3. $C = B - Y_1$

4. $D = X_2 \cdot Z_1$
5. $E = D - X_1$

6. $F = X_1 + D$
7. $G = \left(E^2 \cdot F\right) \cdot Z_1$

8. $H = 2X_1 + D$
9. $I = C \cdot H - Y_1 \cdot E$

10. $X_3 = C^2 - G$

11. $Y_3 = E\left(Z_3 \cdot I - Z_1 \cdot C^3\right)$

12. $Z_3 = A \cdot E^2$
13. $Return(X_3, Y_3, Z_3)$

Finally, we can calculate $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$ where $X_3 = C^2 - G$, $Y_3 = E\left(Z_3 \cdot I - Z_1 \cdot C^3\right)$ and $Z_3 = A \cdot E^2$. Here we can use Urdhva-Tiryagbhyam and Dvandva-Yoga technique to evaluate C^2, $A \cdot E^2$, $Z_3 \cdot I$, $Z_1 \cdot C^3$ *and* $\left(E^2 \cdot F\right) Z_1$.

Case II. The LDPC System with $c = 1$ and $d = 2$, the point doubling of a point P_1 is given by $P_3 = 2P_1 = (X_3, Y_3, Z_3)$ where

$$X_3 = (3X_1^2 + aZ_1^2)^2 - 8X_1 Y_1^2$$

$$Y_3 = 24 X_1 Y_1^3 Z_1 (3X_1^2 + aZ_1^2)$$

$$-2Y_1 Z_1 (3X_1^2 + aZ_1^2)^3 - 16 Y_1^5$$

$$Z_3 = 4Y_1^2 Z_1$$

6.3.2 Algorithm:

Doubling of a point in Lopez-Dahab Projective Coordinate (LDPC) System:

Input: $P_1 = (X_1, Y_1, Z_1)$, a

Output: $P_3 = (X_3, Y_3, Z_3) = 2P_1$

1. $A = 4Y_1^2$

2. $B = 2X_1 \cdot A$
3. $C = 3X_1^2 + a.Z_1^2$

4. $D = B - X_3$
5. $X_3 = C^2 - B$

6. $Y_3 = 2Y_1 \cdot Z_1 \cdot C - Y_1 \cdot A^2$

7. $Z_3 = A \cdot Z_1$

8. $Return(X_3, Y_3, Z_3)$.

Finally, we can calculate $P_3 = (X_3, Y_3, Z_3) = 2P_1$, where,

$X_3 = C^2 - 2X_1 \cdot B$, $Y_3 = Z_1 \cdot A \cdot C \cdot D - Y_1 \cdot B^2$, *and* $Z_3 = Z_1 \cdot B$

Here $A, B, C,$ *and* D can be calculated by Urdhva-Tiryagbhyam and Dvandva-Yoga technique.

7 RESULT ANALYSIS AND COMPARISON

Cryptographic Operations such as point addition, point doubling are done in the form of algorithms to get minimum arithmetic operations in ECC. The tabulated results for point addition and point doubling is shown in table 1 and table 2. The tables 1, 2 give the comparison between arithmetic operations in ECC and arithmetic operations in Vedic Mathematics sutras based ECC for the SPC, JPC, and LDPC system. The comparison is based on the number of multiplications, square, cube, fourth power and fifth power in point doubling and addition in ECC (Elliptic Curve Cryptography) and VECC (Vedic Mathematics based Elliptic Curve Cryptography). The total arithmetic operations are compared in table 1 and 2 and also in figure 3, 4, 5, 6 of the graphs which concludes that the numbers of operations are minimum in table 2 and purposed result in the graphs which are based on VECC. And also the comparison of the results is done by graphically, figure 3, 4, 5, and 6 show comparison between all arithmetic operations in the formulae of cryptographic operations (point doubling and point addition) and all arithmetic operations in the algorithm of point doubling and point addition. Figure 3 shows the Comparison of total arithmetic operation in point addition for SPC, JPC,

and LDPC systems. Figure 4 shows the Comparison of total arithmetic operations in point doubling for SPC, JPC, and LDPC systems, figure 5 shows the Comparing the arithmetic operations in point addition for SPC, JPC, LDPC systems and finally, figure 6 shows Comparing the arithmetic operations in point doubling for SPC, JPC, LDPC system. In table 3, 4, 5, 6, 7, and 8 we observed that running time of Vedic Mathematics based algorithms takes above 70% (app.) less running time than the using conventional methods based algorithms for 4-bits, 8-bits, and 16-bits binary number.

Point Coordinate System	Point addition						Pont doubling					
	(M)	(S)	(C)	(P_4)	(P_5)	(T)	(M)	(S)	(C)	(P_4)	(P_5)	(T)
Standard Projective	39	3	4	0	0	46	10	6	4	3	1	14
Jacobian Projective	14	9	5	0	0	28	7	6	1	4	0	18
Lopez-Dahab Projective	20	9	2	1	0	32	7	8	2	0	1	18

Table 1. Number of operations needed in addition and doubling of points in ECC for SPC, JPC, LDPC System:

Point Coordinate System	Point addition						Pont doubling					
	(M)	(S)	(C)	(P_4)	(P_5)	(T)	M)	(S)	(C)	(P_4)	(P_5)	(T)
Standard Projective	14	2	2	0	0	18	6	3	1	0	0	10
Jacobian Projective	08	3	0	0	0	11	4	4	0	1	0	09
Lopez-Dahab Projective	10	3	1	0	0	14	6	4	0	0	0	10

Table 2. Number of operations needed in addition and doubling of points in VECC for SPC, JPC, LDPC System Using AIVM algorithms:

8 Bits Running Time for Standard Projective coordinate	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.011186 app. (s)	0.0034026 app. (s)	69.5826%
Point Doubling	0.010767 app. (s)	0.0022622 app. (s)	78.9893%

Table 3. Synthesis results of point addition and point doubling

16 Bits Running Time for Standard Projective coordinate	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.011407 app. (s)	0.0019949 app. (s)	82.5111%
Point Doubling	0.010691 app. (s)	0.00261 app. (s)	75.5866%

Table 4. Synthesis results of point addition and point doubling

8 Bits Running Time for Jacobian Projective coordinate system	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.011009 app. (s)	0.0025012 app. (s)	77.2803%
Point Doubling	0.01048 app. (s)	0.0030219 app. (s)	71.1643%

Table 5. Synthesis results of point addition and point doubling

16 Bits Running Time for Jacobian Projective coordinate system	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.010868 app. (s)	0.0020361 app. (s)	81.2646%
Point Doubling	0.011017 app. (s)	0.00856 app. (s)	77.4386%

Table 6. Synthesis results of point addition and point doubling

8 Bits Running Time for Lopez- Dahab Projective	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.011041 app. (s)	0.0022075 app. (s)	80.0051%
Point Doubling	0.01064 app. (s)	0.0026948 app. (s)	74.7301%

Table 7. Synthesis results of point addition and point doubling

16 Bits Running Time for Lopez- Dahab Projective	Running Time Processed using Conventional Methods	Running Time Processed using Vedic Mathematics Sutras	Average of Reducing Time
Point Addition	0.0107 app. (s)	0.0020237 app. (s)	81.7189%
Point Doubling	0.011653 app. (s)	0.0026285 app. (s)	75.3256%

Table 8. Synthesis results of point addition and point doubling

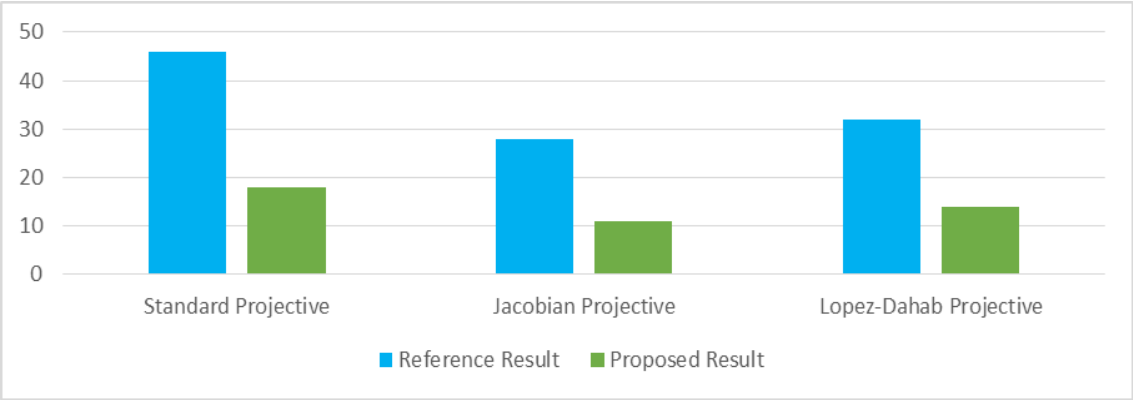

Figure 3. Comparison of total arithmetic operations in point addition for SPC, JPC, LDPC systems

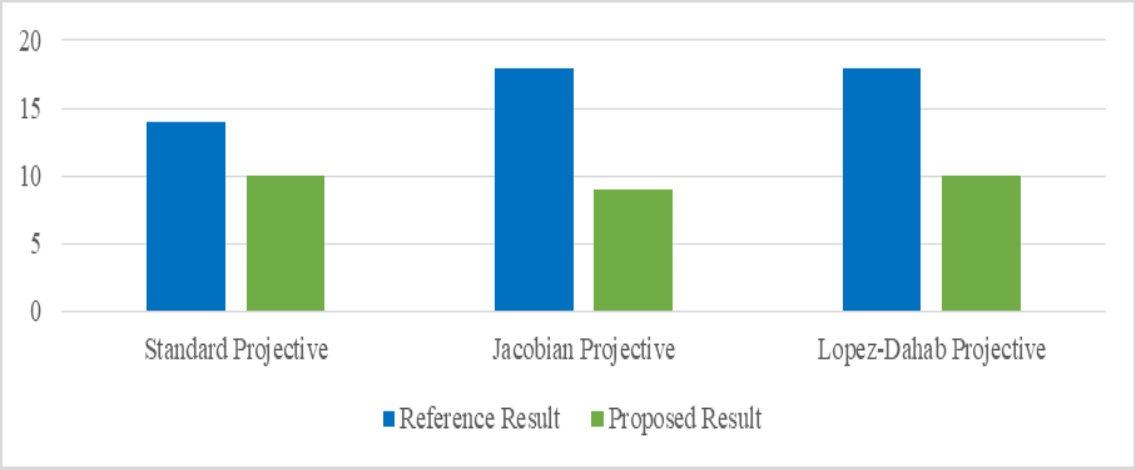

Figure 4. Comparison of total arithmetic operations in point doubling for SPC, JPC, and LDPC systems

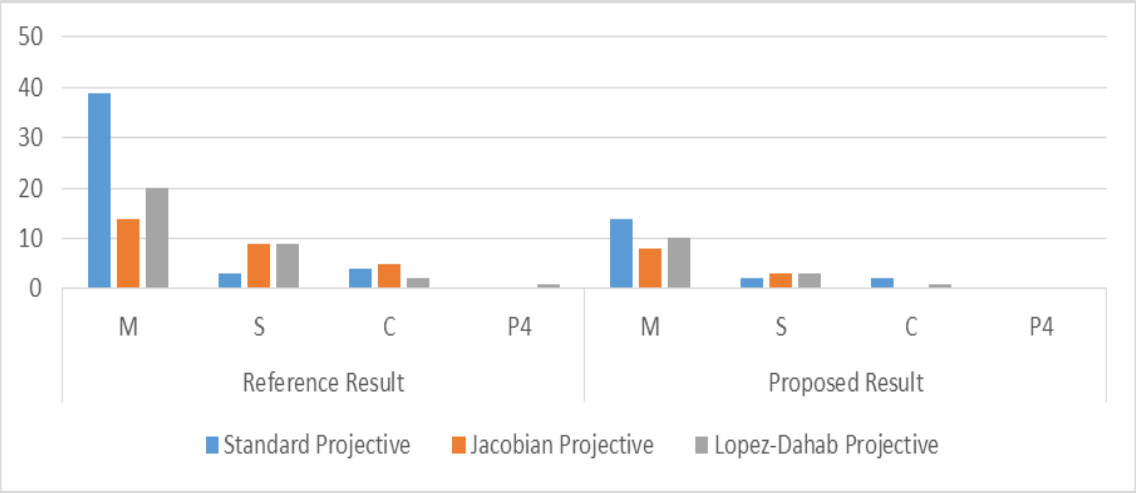

Figure 5. Comparing the arithmetic operations in Point addition for SPC, JPC, LDPC systems

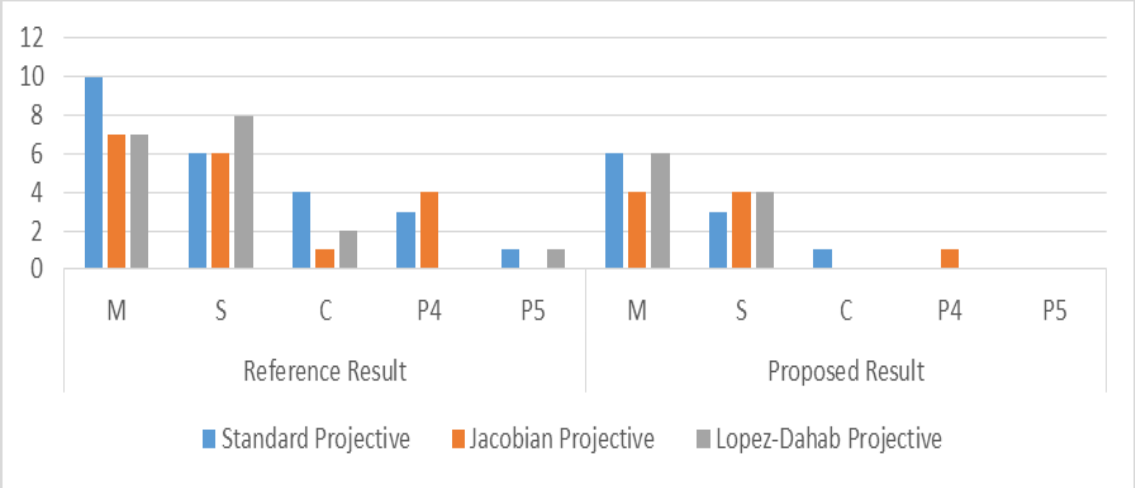

Figure 6. Comparing the arithmetic operations in Point doubling for SPC, JPC, LDPC systems

8 CONCLUSION

We designed a high-performance ECC over the Projective coordinate system using AIVM sutras. There are mainly two useful operations elaborated in ECC first one is the addition and the other is doubling. In these major operations, all multiplications and squaring are calculated by AIVM sutras and algorithms to push forward the all scalar multiplications. Our approach described the squaring and multiplications in point doubling and addition of ECC, this paper shows AIVM sutras based calculations in multiplications and squares give the best performance than the conventional calculations. This approach is effectual in the points of speed, area, and security. Various Ancient Indian Vedic Mathematics sutras are discussed above and these sutras have been implemented in MATLAB Tool and are synthesized and simulated by MATLAB Tool. The code is written for the square of 8-bit, and 16-bit binary number and multiplication code for 8-bit, and 16-bit binary numbers over the projective coordinate systems. For cryptographic operations (point addition, point doubling) using Vedic Mathematics, the Average of Reducing Time is above 70% (appx.) less than conventional methods. Future work of the AIVM based ECC will be improved all cryptographic-based security system, such as, IP Security, e-mail, e-commerce, internet banking, secure communications and information system.

ACKNOWLEDGMENT

We would like to thank Swami Bharati Krishna Tirthaji for the wonderful Ancient Indian Mathematics that he created. I also wish to acknowledge my sincere gratitude to the entire Gurukula Kangri Vishwavidyalaya, Haridwar (Uttarakhand).

REFERENCES

[1] Alkhatib M., Jaafar A., Zukerman Z., and Rushden M., "The Design of Projective Binary Edwards Elliptic Curves over GF (P) benefiting from mapping elliptic curves computation to variable degree of parallel design," *International Journal on Computer Science and Engineering*, vol. 3, no. 4, pp. 1697-1712, 2011.

[2] Anchalya R., Chiranjeevi G. N., and Kulkarni S., "Efficient Computing Techniques using Vedic Mathematics Sutras," *International Journal of Innovative Research in Electrical Electronic Instrumentation and control engineering,* vol. 3 no. 5, pp. 24-27, 2015.

[3] Deepa A., and Marimuthu C. N., "Squaring using Vedic Mathematics and its architectures a survey," *International Journal of Intellectual Advancements and Research in Engineering Computations,* vol. 6, no. 1, pp. 214-218, 2018.

[4] Diffie W., and Hellman M., "New directions in Cryptography," *IEEE Transactions on information Theory,* vol. 22 no. 6, pp. 644-654, 1976.

[5] Gaikwad K. M., and Chavan M. S., "Vedic Mathematics for Digital Signal Processing Operations: A Review," *International Journal of Computer Applications*, vol. 131, no. 8, pp. 10-14, 2015.

[6] Gutub A.A.A., "Remodeling of Elliptic Curve Cryptography Scalar Multiplication Architecture using Parallel Projective Coordinate System," *International Journal of Computer Science and Security,* vol. 4 no. 4, pp. 409-425, 2010.

[7] Hankerson D., Menzes A., and Vanstone S., "*Guide to Elliptic Curve Cryptography*," Springer-Verlag, New York, 2004.

[8] Kan he A., Das S.K., and Singh A.K., "Design and implementation of low power multiplier using Vedic multiplication technique," *International Journal of Computer Science and Communication,* vol. 3, no. 1, pp. 131-132, 2012.

[9] Koblitz N., "Elliptic Curve Cryptosystem," *Journal of Mathematics Computation,* vol. 48 no. 177, pp. 203-209, 1987.

[10] Lisha A., and Monoth T., "Analysis of cryptography algorithms based on Vedic Mathematics," *International Journal of Applied Engineering Research,* vol. 13, no. 3, pp. 68-72, 2018.

[11] Menezes A. J., "*Elliptic Curve Public Key Cryptosystems*," Kluwer Academic Publishers, Springer, 1993.

[12] Nanda A., and Behera S., "Design and Implementation of Urdhva-Tiryagbhyam Based Fast 8×8 Vedic Binary Multiplier," *International Journal of Engineering Research & Technology*, vol. 3, no. 3, pp. 1856-1859, 2014.

[13] Pinagle S. D., "A Survey of Trends in Cryptography and Curve Cryptography," *International Journal of Scientific Research and Education,* vol. 4, no. 50, pp. 5294-5301, 2016.

[14] Poornima M., Patil S. K., Kumar S., Shridhar K. P., and Sanjay H., "Implementation of multiplier using Vedic algorithm. International Journal of Innovative Technology and Exploring Engineering, vol. 2, no. 6, pp. 2278-3075, 2013.

[15] Salim S. M., and Lakhotiya S. A., "Implementation of RSA Cryptosystem using Ancient Indian Vedic Mathematics," *International Journal of Science and Research,* vol. 4 no. 5, pp. 3221-3230, 2015.

[16] Sameer G., Sumana M., and Kumar S., "Novel High Speed Vedic Mathematics Multiplier using Compressors," *International Journal of Advanced Technology and Innovative Research,* vol. 7 no. 2, pp. 0244-0248, 2015.

[17] Shembalkar S., Dhole S., Yadav T., Thakre P., "Vedic Mathematics Sutra- A Review," *International Conference on Recent Trends in Engineering Science and Technology,* vol. 5, no. 1, p. 148-155, 2017.

[18] Shylashree N., Reddy D. V. N., and Sridhar V., "Efficient Implementation of Scalar Multiplication for Elliptic Curve Cryptography using Ancient Indian Vedic Mathematics over GF(p)," *International Journal of Computer Applications,* vol. 49 no. 7, pp. 0975-8887, 2012.

[19] Stallings W., "*Cryptography and Network Security Principals and Practices,*" Prentice Hall, India 2003.

[20] Thapliyal H., and Arbania H.R., "A Time-Area-Power Efficient Multiplier and Square Architecture Based on Ancient Indian Vedic Mathematics," *Proceedings of the 2004 International Conference on VLSI*, Las Vegas, Nevada, pp. 434-439, 2004.

[21] Thomas C., Sheela G., and Saranya P. K., "A Survey on Various Algorithm Used for Elliptic Curve Cryptography," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 6, pp. 7296-7301, 2014.

[22] Tirthaji J. S. S. B. K., "*Vedic Mathematics or Sixteen Simple Sutras from Vedas,*" Motilal Bhandaridas Varanasi India, 1986.

[23] Warang M, and Tambe A., "A review on high speed complex multiplier using Vedic Mathematics: an effective tool," *International Journal of Advance Electrical Electronics Engineering,* vol. 5, no. 1, pp. 26-28, 2016.