INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH AND INNOVATIVE STUDIES
ISSN: 2820-7157
www.ijsrisjournal.com

June 2024 **Volume 3** **Number 3** **Pages 46-54**

Received Date: April 10, 2024 **Accepted Date:** May 11, 2024 **Published Date:** June 01, 2024

Available Online at https://www.ijsrisjournal.com/index.php/ojsfiles/article/view/156

https://doi.org/10.5281/zenodo.11244720

New Compression Point Reducing Memory Size in Field of Characteristic Different From 2 And 3

ASSOUJAA ISMAIL[1], EZZOUAK SIHAM [2]

[1]Sidi Mohammed Ben Abdellah University FSDM (labo: LASMA), Fez; Morocco, ismail.assoujja@usmba.ac.ma
[2]Sidi Mohammed Ben Abdellah University FSDM (labo: LASMA), Fez; Morocco, siham.ezzouak@usmba.ac.ma

ABSTRACT

Compression point is a new method to compress the space memory and still have the same data. In this paper, we will present a new method of compression points work well with addition opera- tion in elliptic curve, so instead of storing the value of two points P = (xP , yP), Q = (xQ , yQ), we will store theaddition of the x- coordinates i,e ($\alpha = xP + xQ , yP , yQ$) or the y-coordinates i,e ($xP , xQ , \beta = yP + yQ$). In this article, we show a new technique for compressing two points in elliptic curve with different coordinate system: Affine, Projective and Jacobian in a field of characteristic ≠ 2 & 3, and show the cost of theses operations. This method can save if we work with affine, Projective or Jacobian coordinates, at least 25%, 17%, 17% of memory size respectively, and also see what happens in case if we take Edwards curve and Montgomery curve cases.

Keywords: Elliptic curve, Affine coordinate, Projective coordinate, Jacobian coordinate, compression point.

1. INTRODUCTION

The first study of Elliptic curve cryptosystems, was startedby N. Koblitz [1] and V. Miller [2] in the 90's (1986-1987), independently re- quires smaller key sizes than the other public cryptosystems. In the 21th centuries, a lot of researchers was developing more and more application in this field ([6], [7], [8], [9], [10], [11]...). Khabbazian et al. in [5] have noticed that any two points on the curve P and Q consisting of the (x_P , y_P) and (x_Q , y_Q) in affine coordinate,those coordinates can be compressed into just $(\alpha = x_P + x_Q , y_P , y_Q)$ or $(x_P , x_Q , \beta = y_P + y_Q)$, this compression can save multiplications and inversions operations. Also, we can do thesame with Jacobian and projective coordinates, so instead of taking $(X_P , X_Q , Y_P , Y_Q , Z_P , Z_Q)$, we only keep $(\alpha = X_P + X_Q , Y_P , Y_Q , Z_P , Z_Q)$ or $(X_P , X_Q , \beta = Y_P + Y_Q , Z_P , Z_Q)$, this compression can save 17% of the memory space. This paper is the enlarge these previous results and here we continue this our investigation by examining what happens when we work in a field of characteristic ≠ 2,3. More precisely, we will compare between our work and the classical addition operation of two points in elliptic curves in a field of characteristic 2,3 with affine, projective and Jacobian coordinates. The paper is organized as follow: Section 2, we recall some background on the basics arithmetic operation of elliptic curves in a field of characteristic ≠ 2, 3 with affine, projective and Jacobian coordinates. Section 3 describes
our main theorems and the results of our work.
Eventually, section 4 concludes the paper

2. MATHEMATICAL BACKGROUND

In everything that follows, we shall use, without explicitmention, the following:

- p: prime number.
- q: power of prime number.
- F: field of characteristic different from 2 and 3.
- P∞: point at infinity.
- Fp: The finite field containing p elements, where pis a prime.
- E: An elliptic curve over the field Fq.
- E (Fq): The set of all points on an elliptic curve Edefined over Fq and including P∞.
- I: inversion.
- M: multiplication.
- S: squaring.
- C: cube.

2.1 Arithmetic of elliptic curve

An elliptic curve is a smooth curve defined by a polynomial equation of degree three. The general form of such a curve defined over a field $\mathbb{F}$ is a set of points whose coordinates satisfy an equation: $f(x, y) = 0$

$$a_1x^3+a_2x^2y+a_3xy^2+a_4y^3+a_5x^2+a_6xy+a_7y^2+a_8x+a_9y+a_{10} = 0,$$

with coefficients a_i in the field $\mathbb{F}$.
At least one of the coefficients a_1, a_2, a_3, or a_4 must be non-zero, to ensure that the polynomial $f(x, y)$ is actually of degree three. ([13] 2-17)

Definition 2.1. (Weierstrass equation:) ([13] 2-18)
An elliptic curve E over a finite field $\mathbb{F}_p$, for p a prime number is the set of points verifying the Weierstrass equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

for $a_i \in \mathbb{F}_p$.
We note that the indexes of the constants a_1, a_3, a_2, a_4, and a_6 are not chosen at random. They are the complement to 6 of the degree of the monomials, counting a degree two, three, and zero for x and y, respectively.

Affine coordinates: ([13] 2-19)
Let $\mathbb{F}$ be a field of characteristic different from 2 and 3. An elliptic curve is a set defined by:

$$E(\mathbb{F}) = \{(x, y) \in \mathbb{F} \| y^2 = x^3 + ax + b\} \cup \{P_\infty\},$$

where $\Delta = 4a^3 + 27b^2$is non-zero.
Projective coordinates: ([13] 2-19)
Let $\mathbb{F}$ be a field of characteristic different from 2 and 3. An elliptic curve is the subset of the projective plane P_2 defined by:

$$E(\mathbb{F}) = \{(X, Y, Z) \in P_2(\mathbb{F}) \| Y^2Z = X^3 + aXZ^2 + bZ^3\} \cup \{P_\infty\},$$

where $\Delta = 4a^3 + 27b^2$is non-zero.
Jacobian coordinates: ([13] 2-22)
Let $\mathbb{F}$ be a field of characteristic different from 2 and 3. In Jacobian coordinates, the elliptic curve E is given by:

$$E(\mathbb{F}) = \{(X, Y, Z) \in \mathbb{F} \| Y^2 = X^3 + aXZ^4 + bZ^6\} \cup \{P_\infty\},$$

where $\Delta = 4a^3 + 27b^2$is non-zero.

- **Addition of two points P and Q $\in$ E($\mathbb{F}_p$),**

We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$.
The table below, show the addition formulas in affine, projective and Jacobian coordinates, also their cost.

Table 1: Addition formulas in Elliptic curve

Coordinate	Addition formula	with
Affine	$x_3 = \lambda^2 - x_1 - x_2$ $y_3 = \lambda(x_1 - x_3) - y_1$	$\lambda = \frac{y_1-y_2}{x_1-x_2}$
Projective	$X_3 = BC$ $Y_3 = A(B^2X_1Z_2 - C) - B^3Y_1Z_2$ $Z_3 = B^3Z_1Z_2$	$A = Y_2Z_1 - Y_1Z_2$ $B = X_2Z_1 - X_1Z_2$ $C = A^2Z_1Z_2 - B^3 - 2B^2X_1Z_2$
Jacobian	$X_3 = -E^3 - 2AE^2 + F^2$ $Y_3 = -CE^3 + F(AE^2 - X_3)$ $Z_3 = Z_1Z_2E$	$A = X_1Z_2^2, B = X_2Z_1^2$ $C = Y_1Z_2^3, D = Y_2Z_1^3$ $E = B - A, F = D - C$

3 COMPRESSION POINT

3.1 First new compression points to Reduce Memory

To work with these methods of compression point, we follow these steps:

(1) Choose which coordinate system we will work with (i,e affine, projective or Jacobian Coordinates) and the equation link it with this system coordinate.
(2) In affine coordinate, instead of store the values of $P = (x_1, y_1)$ and
$Q = (x_2, y_2)$, we put only the value of $(x_1, x_2, \alpha = y_1 + y_2)$.
- In projective or Jacobian coordinates, instead of store the values of
$P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, we put only the value of $(X_1, X_2, \alpha = Y_1 + Y_2, Z_1, Z_2)$.
(3) Calculate the value of $\beta = y_1 - y_2$.
(4) Get the value of y_1 and y_2 from α and β.
(5) Replace the value of y_1, y_2 in the formula of addition from table 1 above to get (x_3, y_3).

Affine coordinates:
In affine coordinates, the elliptic curve equation of E is given by:

$$y^2 = f(x) = x^3 + ax + b.$$

Instead of take the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$
we keep only the value of $(x_1, x_2, \alpha = y_1 + y_2)$.
We can calculate

$$\beta = y_1 - y_2 = \frac{(y_1 - y_2)(y_1 + y_2)}{y_1 + y_2} = \frac{y_1^2 - y_2^2}{y_1 + y_2} = \frac{f(x_1) - f(x_2)}{\alpha}$$

so $y_1 = \frac{\alpha+\beta}{2}$ and $y_2 = \frac{\alpha-\beta}{2}$
To compute β we need I+2C=I+2M+2S.

Algorithm 1 y-coordinate decompression with affine coordinate

Input: $(x_1, x_2, \alpha = y_1 + y_2)$
Output: (y_1, y_2)
1: $\beta \leftarrow y_1 + y_2 = \frac{x_1^3-x_2^3+a(x_1-x_2)}{\alpha}$
2: $y_1 \leftarrow \frac{\alpha+\beta}{2}$
3: $y_2 \leftarrow \frac{\alpha-\beta}{2}$
4: return (y_1, y_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$.
We replace the value of y_1 and y_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 2I+4M+3S.

Projective coordinates:

In projective coordinates, the elliptic curve equation of E is given by:

$$Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Instead of take the values of $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$, we put only the value of $(X_1, X_2, \alpha = Y_1 + Y_2, Z_1, Z_2)$.

We have, $Y_1^2Z_1 = X_1^3 + aX_1Z_1^2 + bZ_1^3$ and $Y_2^2Z_2 = X_2^3 + aX_2Z_2^2 + bZ_2^3$

So, $Y_1^2Z_1Z_2 = (X_1^3 + aX_1Z_1^2 + bZ_1^3)Z_2$ and $Y_2^2Z_2Z_1 = (X_2^3 + aX_2Z_2^2 + bZ_2^3)Z_1$

We can calculate

$$\beta = Y_1 - Y_2 = \frac{(Y_1 - Y_2)(Y_1 + Y_2)}{Y_1 + Y_2} = \frac{Y_1^2 - Y_2^2}{Y_1 + Y_2} = \frac{(Y_1^2 - Y_2^2)Z_1Z_2}{(Y_1 + Y_2)Z_1Z_2}$$

$$\beta = \frac{Y_1^2Z_1Z_2 - Y_2^2Z_1Z_2}{\alpha Z_1Z_2} = \frac{(X_1^3 + aX_1Z_1^2 + bZ_1^3)Z_2 - (X_2^3 + aX_2Z_2^2 + bZ_2^3)Z_1}{\alpha Z_1Z_2}$$

so

$$Y_1 = \frac{\alpha + \beta}{2} \text{ and } Y_2 = \frac{\alpha - \beta}{2}$$

To compute β we need I+10M+6S.

Algorithm 2 y-coordinate decompression with projective coordinate

Input: $(X_1, X_2, \alpha = Y_1 + Y_2, Z_1, Z_2)$
Output: (Y_1, Y_2)
1: $\beta \leftarrow Y_1 + Y_2 = \frac{(X_1^3 + aX_1Z_1^2 + bZ_1^3)Z_2 - (X_2^3 + aX_2Z_2^2 + bZ_2^3)Z_1}{\alpha Z_1Z_2}$
2: $Y_1 \leftarrow \frac{\alpha + \beta}{2}$
3: $Y_2 \leftarrow \frac{\alpha - \beta}{2}$
4: return (Y_1, Y_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points
$P \oplus Q = (X_3, Y_3, Z_3)$.
We replace the value of Y_1 and Y_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires I+22M+8S.

Jacobian coordinates:

In Jacobian coordinates, the elliptic curve equation of E is given by:

$$Y^2 = f(X, Z) = X^3 + aXZ^4 + bZ^6.$$

Instead of take the values of $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$ we only keep the value of $(X_1, X_2, \alpha = Y_1 + Y_2, Z_1, Z_2)$.
We can calculate

$$\beta = Y_1 - Y_2 = \frac{(Y_1 - Y_2)(Y_1 + Y_2)}{Y_1 + Y_2} = \frac{Y_1^2 - Y_2^2}{Y_1 + Y_2} = \frac{f(X_1, Z_1) - f(X_2, Z_2)}{\alpha}$$

so

$$Y_1 = \frac{\alpha + \beta}{2} \text{ and } Y_2 = \frac{\alpha - \beta}{2}$$

To compute β we need I+6M+8S.

Algorithm 3 y-coordinate decompression with Jacobian coordinate

Input: $(X_1, X_2, \alpha = Y_1 + Y_2, Z_1, Z_2)$
Output: (Y_1, Y_2)
1: $\beta \leftarrow Y_1 + Y_2 = \frac{(X_1^3 + aX_1Z_1^4 + bZ_1^6) - (X_2^3 + aX_2Z_2^4 + bZ_2^6)}{\alpha}$
2: $Y_1 \leftarrow \frac{\alpha + \beta}{2}$
3: $Y_2 \leftarrow \frac{\alpha - \beta}{2}$
4: return (Y_1, Y_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points
$P \oplus Q = (X_3, Y_3, Z_3)$.
We replace the value of Y_1 and Y_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires I+18M+12S.

Comparison:

The table below show the cost of decompression and addition formulas of two points with affine, projective and Jacobian coordinates

Table 2: The cost of classical addition decompression and new addition

Coordinate	Classical addition	Decompression	New Addition	Saving
Affine	I+2M+S	1I+2M+2S	2I+4M+3S	25%
Projective	12M+2S	I+10M+6S	I+22M+8S	17%
Jacobian	12M+4S	I+6M+8S	I+18M+12S	17%

- In the table above we see that the cost of addition in affine coordinate change from I+2M+S to 2I+4M+3S to save 25% of memory.
- In projective coordination the cost of addition change from 12M+2S to I+22M+8S to save 17% of memory.
- Finally, in Jacobian coordinate the cost of addition change from 12M+4S to I+18M+12S to save also 17% of memory.

3.2 Second new compression points to Reduce Memory

To work with these methods of compression point, we follow these steps:

(1) Choose which coordinate system we will work with (i,e affine, projective or Jacobian Coordinates) and the equation link it with this system coordinate.
(2) In affine coordinate, instead of store the values of $P = (x_1, y_1)$ and
$Q = (x_2, y_2)$, we put only the value of $(\alpha = x_1 + x_2, y_1, y_2)$.
- In projective coordinates, instead of store the values of $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, we put only the value of $(\alpha = \frac{X_1}{Z_1} + \frac{X_2}{Z_2}, Y_1, Y_2, Z_1, Z_2)$.
- In Jacobian coordinates, instead of store the values of $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, we put only the value of $(\alpha = \frac{X_1}{Z_1^2} + \frac{X_2}{Z_2^2}, Y_1, Y_2, Z_1, Z_2)$.
(3) Do the calculation to find the value of x_1 and x_2.
(4) Replace the value of x_1, x_2 in the formula of addition from table 1 above to get (x_3, y_3).

Affine coordinates:
In affine coordinates, the elliptic curve equation of E is given by:

$$y^2 = f(x) = x^3 + ax + b.$$

Instead of take the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$
we keep only the value of $(\alpha = x_1 + x_2, y_1, y_2)$.
We have

$$y_1^2 + y_2^2 = x_1^3 + x_2^3 + a(x_1 + x_2) + 2b$$

So $y_1^2+y_2^2 = (x_1+x_2)(x_1^2-x_1x_2+x_2^2+a)+2b = \alpha(x_1^2-x_1x_2+x_2^2+a)+2b$

Hence

$$\beta = x_1^2 - x_1x_2 + x_2^2 = \frac{y_1^2 + y_2^2 - 2b}{\alpha} - a$$

Also

$$x_1x_2 = \frac{\alpha^2 - \beta}{3}$$

We can calculate

$$\gamma = x_1^2 + x_1x_2 + x_2^2 = \beta + \frac{2}{3}(\alpha^2 - \beta)$$

We have also

$$\text{We have also } y_1^2 - y_2^2 = x_1^3 - x_2^3 + a(x_1 - x_2)$$

So

$$y_1^2 - y_2^2 = (x_1 - x_2)(x_1^2 + x_1x_2 + x_2^2 + a) = (x_1 - x_2)(\gamma + a)$$

We obtain

$$\delta = x_1 - x_2 = \frac{y_1^2 - y_2^2}{\gamma + a}$$

Hence

$$x_1 = \frac{\alpha + \delta}{2} \text{ and } x_2 = \frac{\alpha - \delta}{2}$$

To compute x_1 and x_2 we need 2I+3S.

Algorithm 4 x-coordinate decompression with affine coordinate

Input: $(\alpha = x_1 + x_2, y_1, y_2)$
Output: (x_1, x_2)
1: $\beta \leftarrow x_1^2 - x_1x_2 + x_2^2 = \frac{y_1^2+y_2^2-2b}{\alpha} - a$
2: $x_1x_2 \leftarrow \frac{\alpha^2-\beta}{3}$
3: $\gamma \leftarrow x_1^2 + x_1x_2 + x_2^2 = \beta + \frac{2}{3}(\alpha^2 - \beta)$
4: $\delta \leftarrow x_1 - x_2 = \frac{y_1^2-y_2^2}{\gamma+a}$
5: $x_1 \leftarrow \frac{\alpha+\delta}{2}$
6: $x_2 \leftarrow \frac{\alpha-\delta}{2}$
7: return (x_1, x_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$.
We replace the value of x_1 and x_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I+2M+4S.

Projective coordinates:
In projective coordinates, the elliptic curve equation of E is given by:

$$Y^2Z = X^3 + aXZ^2 + bZ^3.$$

Instead of take the values of $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$, we put only the value of $(\alpha = \frac{X_1}{Z_1} + \frac{X_2}{Z_2}, Y_1, Y_2, Z_1, Z_2)$.
To compress $\alpha = \frac{X_1Z_2+X_2Z_1}{Z_1Z_2}$ we need I+3M.
To decompress α to obtain X_1 and X_2.
First we calculate the value of $\beta = \frac{X_1^2}{Z_1^2} - \frac{X_1X_2}{Z_1Z_2} + \frac{X_2^2}{Z_2^2}$ by

$$\frac{Y_1^2}{Z_1^2} + \frac{Y_2^2}{Z_2^2} = \frac{X_1^3}{Z_1^3} + \frac{X_2^3}{Z_2^3} + \frac{aX_1}{Z_1} + \frac{aX_2}{Z_2} + 2b$$

We arrive at

$$\frac{Y_1^2}{Z_1^2} + \frac{Y_2^2}{Z_2^2} = \beta(\alpha + a) + 2b$$

So

$$\beta = \frac{\frac{Y_1^2}{Z_1^2} + \frac{Y_2^2}{Z_2^2} - 2b}{\alpha} - a$$

After this we calculate the value of $\frac{X_1X_2}{Z_1Z_2}$,
Easy to find it just calculate

$$\frac{X_1X_2}{Z_1Z_2} = \frac{\alpha^2 - \beta}{3}$$

Next we calculate the value of $\gamma = \frac{X_1^2}{Z_1^2} + \frac{X_1X_2}{Z_1Z_2} + \frac{X_2^2}{Z_2^2}$ by

$$\gamma = \beta + \frac{2}{3}(\alpha^2 - \beta)$$

We have also

$$\frac{Y_1^2}{Z_1^2} - \frac{Y_2^2}{Z_2^2} = \frac{X_1^3}{Z_1^3} - \frac{X_2^3}{Z_2^3} + \frac{aX_1}{Z_1} - \frac{aX_2}{Z_2}$$

So

$$\frac{Y_1^2}{Z_1^2} - \frac{Y_2^2}{Z_2^2} = (\frac{X_1}{Z_1} - \frac{X_2}{Z_2})(\beta + a)$$

Hence

$$\delta = \frac{X_1}{Z_1} - \frac{X_2}{Z_2} = \frac{\frac{Y_1^2}{Z_1^2} - \frac{Y_2^2}{Z_2^2}}{\beta} - a$$

So $X_1 = \frac{\alpha+\delta}{2}.Z_1$ and $X_2 = \frac{\alpha-\delta}{2}.Z_2$
To compute X_1 and X_2 we need 2I+7M+5S.

Algorithm 5 x-coordinate decompression with projective coordinate

Input: $(\alpha = \frac{X_1}{Z_1} + \frac{X_2}{Z_2}, Y_1, Y_2, Z_1, Z_2)$
Output: (X_1, X_2)

1: $\beta \leftarrow \frac{X_1^2}{Z_1^2} - \frac{X_1X_2}{Z_1Z_2} + \frac{X_2^2}{Z_2^2} = \frac{\frac{Y_1^2}{Z_1^2} + \frac{Y_2^2}{Z_2^2} - 2b}{\alpha} - a$
2: $\frac{X_1X_2}{Z_1Z_2} \leftarrow \frac{\alpha^2-\beta}{3}$
3: $\gamma \leftarrow \frac{X_1^2}{Z_1^2} + \frac{X_1X_2}{Z_1Z_2} + \frac{X_2^2}{Z_2^2} = \beta + \frac{2}{3}(\alpha^2 - \beta)$
4: $\delta \leftarrow \frac{X_1}{Z_1} - \frac{X_2}{Z_2} = \frac{\frac{Y_1^2}{Z_1^2} - \frac{Y_2^2}{Z_2^2}}{\beta} - a$
5: $X_1 \leftarrow \frac{\alpha+\delta}{2}.Z_1$
6: $X_2 \leftarrow \frac{\alpha-\delta}{2}.Z_2$
7: return (X_1, X_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points
$P \oplus Q = (X_3, Y_3, Z_3)$.
We replace the value of X_1 and X_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I+22M+7S.

Jacobian coordinates:

In Jacobian coordinates, the elliptic curve equation of E is given by:

$$Y^2 = f(X, Z) = X^3 + aXZ^4 + bZ^6.$$

Instead of take the values of $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$, we only keep the value of $(\alpha = \frac{X_1}{Z_1^2} + \frac{X_2}{Z_2^2}, Y_1, Y_2, Z_1, Z_2)$.
To compress $\alpha = \frac{X_1Z_2^2 + X_2Z_1^2}{Z_1^2Z_2^2}$ we need I+3M+2S.
To decompress α to obtain X_1 and X_2.
First we calculate the value of $\beta = \frac{X_1^2}{Z_1^4} - \frac{X_1X_2}{Z_1^2Z_2^2} + \frac{X_2^2}{Z_2^4}$ by

$$\frac{Y_1^2}{Z_1^6} + \frac{Y_2^2}{Z_2^6} = \frac{X_1^3}{Z_1^6} + \frac{X_2^3}{Z_2^6} + \frac{aX_1}{Z_1^2} + \frac{aX_2}{Z_2^2} + 2b$$

We can obviously get $\frac{Y_1^2}{Z_1^6} + \frac{Y_2^2}{Z_2^6} = \alpha(\beta + a) + 2b$
So

$$\beta = \frac{\frac{Y_1^2}{Z_1^6} + \frac{Y_2^2}{Z_2^6} - 2b}{\alpha} - a$$

After this we calculate the value of $\frac{X_1X_2}{Z_1^2Z_2^2}$,
Easy to find it just calculate

$$\frac{X_1X_2}{Z_1^2Z_2^2} = \frac{\alpha^2 - \beta}{3}$$

Next we calculate the value of $\gamma = \frac{X_1^2}{Z_1^4} + \frac{X_1X_2}{Z_1^2Z_2^2} + \frac{X_2^2}{Z_2^4}$, which is

$$\gamma = \beta + \frac{2}{3}(\alpha^2 - \beta)$$

We have also

$$\frac{Y_1^2}{Z_1^6} - \frac{Y_2^2}{Z_2^6} = \frac{X_1^3}{Z_1^6} - \frac{X_2^3}{Z_2^6} + \frac{aX_1}{Z_1^2} - \frac{aX_2}{Z_2^2}$$
$$= (\frac{X_1}{Z_1^2} + \frac{X_2}{Z_2^2})(\gamma + a)$$

Hence

$$\delta = \frac{\frac{Y_1^2}{Z_1^6} - \frac{Y_2^2}{Z_2^6}}{\gamma} - a$$

So $X_1 = \frac{\alpha+\delta}{2}.Z_1^2$ and $X_2 = \frac{\alpha-\delta}{2}.Z_2^2$
To compute X_1 and X_2 we need 2I+9S+7S.

Algorithm 6 x-coordinate decompression with Jacobian coordinate

Input: $(\alpha = \frac{X_1}{Z_1^2} + \frac{X_2}{Z_2^2}, Y_1, Y_2, Z_1, Z_2)$
Output: (X_1, X_2)

1: $\beta \leftarrow \frac{X_1^2}{Z_1^4} - \frac{X_1X_2}{Z_1^2Z_2^2} + \frac{X_2^2}{Z_2^4} = \frac{\frac{Y_1^2}{Z_1^6} + \frac{Y_2^2}{Z_2^6} - 2b}{\alpha} - a$
2: $\frac{X_1X_2}{Z_1^2Z_2^2} \leftarrow \frac{\alpha^2-\beta}{3}$
3: $\gamma \leftarrow \frac{X_1^2}{Z_1^4} + \frac{X_1X_2}{Z_1^2Z_2^2} + \frac{X_2^2}{Z_2^4} = \beta + \frac{2}{3}(\alpha^2 - \beta)$
4: $\delta \leftarrow \frac{X_1}{Z_1^2} - \frac{X_2}{Z_2^2} = \frac{\frac{Y_1^2}{Z_1^6} - \frac{Y_2^2}{Z_2^6}}{\gamma} - a$
5: $X_1 \leftarrow \frac{\alpha+\delta}{2}.Z_1^2$
6: $X_2 \leftarrow \frac{\alpha-\delta}{2}.Z_2^2$
7: return (X_1, X_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (X_1, Y_1, Z_1), Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points
$P \oplus Q = (X_3, Y_3, Z_3)$.
We replace the value of X_1 and X_2 in the addition formula from table 1 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I+24M+13S.

COMPARISON:

The table below show the cost of compression, decompression and addition formulas of two points with affine, projective and Jacobian coordinates

Table 3: The cost of Compression, Decompression and Addition

Coordinate	Cllasical addition	Compression	Decompression	New Addition	Memory saving
Affine	I+2M+S	0	2I+3S	3I+2M+4S	25%
Projective	12M+2S	I+3M	2I+7M+5S	3I+22M+7S	17%
Jacobian	12M+4S	I+3M+2S	2I+9M+7S	3I+24M+13S	17%

- In the table above we see that the cost of addition in affine coordinate change from I+2M+S to 3I+2M+4S to save 25% of memory.
- Also, with projective coordination the cost of addition change from 12M+2S to 3I+22M+7S to save 17% of memory.
- Finally with Jacobian coordinate the cost of addition change from 12M+4S to 3I+24M+13S to save also 17% of memory.

3.3 New compression points in Edwards curves

Harold M. Edwards introduced a new formula for elliptic curves over fields of characteristic $\neq 2$ ([14]). Edwards curve have played a big roles in recent elliptic curve method applications ([15]), technically, it's not an elliptic curve due to its singularities. Even so, we will work with these kind of curve because every Edwards curve is bi-rationally equivalent to an elliptic curve in Weierstrass form, due to symmetry of the Edwards curve.

Affine coordinates:

We can assume that E is given by:

$$y^2 + x^2 = c^2(1 + dx^2y^2) \in \mathbb{F}_p.$$

With $c, d \neq 0$ and $dc^4 \neq 1$

And they also proved that all curves of this form are isomorphic to curve of the form:

$$x^2 + y^2 = 1 + dx^2y^2$$

If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ then

$$P + Q = \left(\frac{x_Py_Q + y_Px_Q}{1 + dx_Px_Qy_Py_Q}, \frac{y_Py_Q - x_Px_Q}{1 - dx_Px_Qy_Py_Q}\right) \quad (1)$$

For these formula, one can easily find that the cost of an addition requires 2I+5M.

Compression point in Edwards curve

To work with these methods of compression point, we follow these steps:

(1) Instead of store the values of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, we put only the value of $(x_1, x_2, \alpha = y_1 + y_2)$ or $(\alpha = x_1 + x_2, y_1, y_2)$.
(2) Calculate the value of $\beta = y_1 - y_2$ or $\beta = x_1 - x_2$.
(3) Get the value of y_1, y_2, x_1, x_2 from α and β.
(4) Replace the value of y_1, y_2, x_1, x_2 in the formula of addition 1 above to get (x_3, y_3).

y or x Coordinate Compression

Instead of take the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$

we only keep the value of $(x_1, x_2, \alpha = y_1 + y_2)$ or $(\alpha' = x_1 + x_2, y_1, y_2)$.

• We have

$$y^2 + x^2 = 1 + dx^2y^2$$

We can calculate

$$y^2 = f(x) = \frac{1 - x^2}{1 - dx^2} \text{ or } x^2 = f(y) = \frac{1 - y^2}{1 - dy^2}$$

So $\beta = y_1 - y_2 = \frac{f(x_1) - f(x_2)}{y_1 + y_2} = \frac{f(x_1) - f(x_2)}{\alpha} = \frac{(1-x_1^2)(1-dx_2^2) - (1-x_2^2)(1-dx_1^2)}{\alpha(1-dx_1^2)(1-dx_2^2)}$

or $\beta' = x_1 - x_2 = \frac{f(y_1) - f(y_2)}{x_1 + x_2} = \frac{f(y_1) - f(y_2)}{\alpha'} = \frac{(1-y_1^2)(1-dy_2^2) - (1-y_2^2)(1-dy_1^2)}{\alpha'(1-dy_1^2)(1-dy_2^2)}$.

The value of $y_1 = \frac{\alpha + \beta}{2}$, $y_2 = \frac{\alpha - \beta}{2}$, $x_1 = \frac{\alpha' + \beta'}{2}$, $x_2 = \frac{\alpha' - \beta'}{2}$

The cost of decompression is 1I+4M+2S.

Algorithm 7 x-coordinate decompression in Edwards curves

Input: $(\alpha = x_1 + x_2, y_1, y_2)$
Output: (x_1, x_2)
1: $\beta \leftarrow x_1 - x_2 = \frac{(1-y_1^2)(1-dy_2^2) - (1-y_2^2)(1-dy_1^2)}{\alpha'(1-dy_1^2)(1-dy_2^2)}$
2: $x_1 \leftarrow \frac{\alpha + \beta}{2}$
3: $x_2 \leftarrow \frac{\alpha - \beta}{2}$
4: return (x_1, x_2)

Algorithm 8 y-coordinate decompression in Edwards curves

Input: $(x_1, x_2, \alpha = y_1 + y_2)$
Output: (y_1, y_2)
1: $\beta \leftarrow y_1 + y_2 = \frac{(1-x_1^2)(1-dx_2^2) - (1-x_2^2)(1-dx_1^2)}{\alpha(1-dx_1^2)(1-dx_2^2)}$
2: $y_1 \leftarrow \frac{\alpha + \beta}{2}$
3: $y_2 \leftarrow \frac{\alpha - \beta}{2}$
4: return (y_1, y_2)

Addition: $P \oplus Q = (x_3, y_3)$

We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$.

We replace the value of y_1 and y_2 in the addition formula 1.

So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I + 9M+ 2S.

Table 4: Decompression, Addition cost in Edwards curve

coordinate	Classical addition	Decompression	New Addition
x or y-coordinate	2I+5M	I+4M+2S	3I+9M+2S

In the table above, we see that the cost of Decompression and Addition of two points in Edwards curve is the same in the case of x-coordinate and y-coordinate compression, due to the symmetry of Edwards curve.

3.4 New compression points in Montgomery curves

Montgomery, founded a new elliptic-curve used in factorization algorithm, this curves and the algorithm associated it, have become very important in the implementation of ECC.

Affine coordinates: ([?])

A Montgomery curve over $\mathbb{F}_q$ is an elliptic curve defined by an affine equation:

$$By^2 = x^3 + ax^2 + x$$

where a and B are parameters in $\mathbb{F}_q$ satisfying $B \neq 0$ and $a^2 \neq 4$.

Addition formula in Montgomery curve

We will show the addition and doubling formulas in the table below:

Table 5: Addition formulas in Montgomery curve

Operation	Formula	with λ
Addition (2.1)	$x_3 = B\lambda^2 - (x_1 + x_2) - a$ $y_3 = \lambda(x_1 - x_3) - y_1$	$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$

We can easily find that the cost of an addition is I+2M+S.

Compression point in Montgomery curve

To work with these methods of compression point, we follow these steps:

(1) Instead of store the values of $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, we put only the value of $(x_1, x_2, \alpha = y_1 + y_2)$ or $(\alpha = x_1 + x_2, y_1, y_2)$.
(2) Calculate the value of $\beta = y_1 - y_2$ or $\delta = x_1 - x_2$.
(3) Get the value of y_1, y_2, x_1, x_2 from α and β or α and δ.
(4) Replace the value of y_1, y_2, x_1, x_2 in the formula of addition from table 5 above to get (x_3, y_3).

y-coordinate compression in Montogomery curve

The Montgomery curve equation of E is given by:

$$By^2 = f(x) = x^3 + ax^2 + x.$$

Instead of storing the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$
we keep only the value of $(x_1, x_2, \alpha = y_1 + y_2)$.
We can calculate

$$\beta = y_1 - y_2 = \frac{(y_1 - y_2)(y_1 + y_2)}{y_1 + y_2} = \frac{y_1^2 - y_2^2}{y_1 + y_2} = \frac{f(x_1) - f(x_2)}{B.\alpha}$$

so $y_1 = \frac{\alpha+\beta}{2}$ and $y_2 = \frac{\alpha-\beta}{2}$
To compute β we need I+2C=I+2M+2S.

Algorithm 9 y-coordinate decompression with affine coordinate

Input: $(x_1, x_2, \alpha = y_1 + y_2)$
Output: (y_1, y_2)
1: $\beta \leftarrow y_1 - y_2 = \frac{x_1^3 - x_2^3 + a(x_1^2 - x_2^2) + x_1 - x_2}{B.\alpha}$
2: $y_1 \leftarrow \frac{\alpha+\beta}{2}$
3: $y_2 \leftarrow \frac{\alpha-\beta}{2}$
4: return (y_1, y_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,

We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$.
We replace the value of y_1 and y_2 in the addition formula from table 5 above.
So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 2I+4M+3S.

x-coordinate compression in Montogomery curve with a=0

The Montgomery curve equation of E is given by:

$$By^2 = f(x) = x^3 + x, \text{ with } a = 0$$

Instead of take the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$
we keep only the value of $(\alpha = x_1 + x_2, y_1, y_2)$.
We have

$$B.(y_1^2 + y_2^2) = x_1^3 + x_2^3 + (x_1 + x_2)$$

So $B.(y_1^2+y_2^2) = (x_1+x_2)(x_1^2-x_1x_2+x_2^2+1) = \alpha(x_1^2-x_1x_2+x_2^2+1)$

Hence

$$\beta = x_1^2 - x_1x_2 + x_2^2 = \frac{B.(y_1^2 + y_2^2)}{\alpha} - 1$$

Also

$$x_1x_2 = \frac{\alpha^2 - \beta}{3}$$

We can calculate

$$\gamma = x_1^2 + x_1x_2 + x_2^2 = \beta + \frac{2}{3}(\alpha^2 - \beta) = \frac{2\alpha^2 + \beta}{3}$$

We have also

$$B.(y_1^2 - y_2^2) = x_1^3 - x_2^3 + (x_1 - x_2)$$

So

$$B.(y_1^2 - y_2^2) = (x_1 - x_2)(x_1^2 + x_1x_2 + x_2^2 + 1) = (x_1 - x_2)(\gamma + 1)$$

We obtain

$$\delta = x_1 - x_2 = \frac{B.(y_1^2 - y_2^2)}{\gamma + 1}$$

Hence

$$x_1 = \frac{\alpha + \delta}{2} \text{ and } x_2 = \frac{\alpha - \delta}{2}$$

To compute x_1 and x_2 we need 2I+3S.

Algorithm 10 x-coordinate decompression in M.C curve with a=0

Input: $(\alpha = x_1 + x_2, y_1, y_2)$
Output: (x_1, x_2)

1: $\beta \leftarrow x_1^2 - x_1x_2 + x_2^2 = \frac{B.(y_1^2+y_2^2)}{\alpha} - 1$
2: $x_1x_2 \leftarrow \frac{\alpha^2-\beta}{3}$
3: $\gamma \leftarrow x_1^2 + x_1x_2 + x_2^2 = \frac{2\alpha^2+\beta}{3}$
4: $\delta \leftarrow x_1 - x_2 = \frac{B.(y_1^2-y_2^2)}{\gamma+1}$
5: $x_1 \leftarrow \frac{\alpha+\delta}{2}$
6: $x_2 \leftarrow \frac{\alpha-\delta}{2}$
7: return (x_1, x_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$. We replace the value of x_1 and x_2 in the addition formula from table 5 above. So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I+2M+4S.

x-coordinate compression in Montogomery curve with $a \neq 0$

The Montgomery curve equation of E is given by:

$$By^2 = f(x) = x^3 + ax^2 + x$$

Instead of take the values of $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$, we keep only the value of $(\alpha = x_1 + x_2, y_1, y_2)$.
We have

$$\begin{aligned} B.(y_1^2 + y_2^2) &= x_1^3 + x_2^3 + a(x_1^2 + x_2^2) + (x_1 + x_2) \\ &= (x_1 + \frac{a}{3})^3 + (x_2 + \frac{a}{3})^3 + (x_1 + x_2)(1 - \frac{a^2}{3}) - \frac{2a^3}{27} \\ &= (x_1 + x_2 + \frac{2a}{3})(x_1^2 - x_1x_2 + x_2^2 + a(x_1 + x_2) + \frac{a^2}{3}) \\ &\quad + (x_1 + x_2)(1 - \frac{a^2}{3}) - \frac{2a^3}{27} \end{aligned}$$

So $\beta = x_1^2 - x_1x_2 + x_2^2$

$$= \frac{B.(y_1^2 + y_2^2) - (x_1 + x_2)(1 - \frac{a^2}{3}) + \frac{2a^3}{27}}{x_1 + x_2 + \frac{2a}{3}} - a(x_1 + x_2) - \frac{a^2}{3}$$

Hence

$$\beta = x_1^2 - x_1x_2 + x_2^2 = \frac{B.(y_1^2 + y_2^2) - \alpha(1 - \frac{a^2}{3}) + \frac{2a^3}{27}}{\alpha + \frac{2a}{3}} - a.\alpha - \frac{a^2}{3}$$

Also $x_1x_2 = \frac{\alpha^2-\beta}{3}$
We can calculate

$$\gamma = x_1^2 + x_1x_2 + x_2^2 = \beta + \frac{2}{3}(\alpha^2 - \beta) = \frac{2\alpha^2 + \beta}{3}$$

We have also

$$\begin{aligned} B.(y_1^2 - y_2^2) &= x_1^3 - x_2^3 + a(x_1 - x_2)^2 + (x_1 - x_2) \\ &= (x_1 - x_2)(x_1^2 + x_1x_2 + x_2^2 + a.\alpha + 1) = (x_1 - x_2)(\gamma + a.\alpha + 1) \end{aligned}$$

We obtain $\delta = x_1 - x_2 = \frac{B.(y_1^2-y_2^2)}{\gamma+a.\alpha+1}$
Hence $x_1 = \frac{\alpha+\delta}{2}$ and $x_2 = \frac{\alpha-\delta}{2}$
To compute x_1 and x_2 we need 2I+3S.

Algorithm 11 x-coordinate decompression in M.C with $a \neq 0$

Input: $(\alpha = x_1 + x_2, y_1, y_2)$
Output: (x_1, x_2)

1: $\beta \leftarrow x_1^2 - x_1x_2 + x_2^2 = \frac{B.(y_1^2+y_2^2)-\alpha(1-\frac{a^2}{3})+\frac{2a^3}{27}}{\alpha+\frac{2a}{3}} - a.\alpha - \frac{a^2}{3}$
2: $x_1x_2 \leftarrow \frac{\alpha^2-\beta}{3}$
3: $\gamma \leftarrow x_1^2 + x_1x_2 + x_2^2 = \frac{2\alpha^2+\beta}{3}$
4: $\delta \leftarrow x_1 - x_2 = \frac{B.(y_1^2-y_2^2)}{\gamma+a.\alpha+1}$
5: $x_1 \leftarrow \frac{\alpha+\delta}{2}$
6: $x_2 \leftarrow \frac{\alpha-\delta}{2}$
7: return (x_1, x_2)

• **Addition:** Addition of two points P and $Q \in E(\mathbb{F}_p)$,
We will take two points $P = (x_1, y_1), Q = (x_2, y_2)$ such that $P \neq \pm Q$ and give the formulas to calculate the addition of these two points $P \oplus Q = (x_3, y_3)$. We replace the value of x_1 and x_2 in the addition formula from table 5 above. So if we want to calculate the cost of this new addition formulas, we can just add the cost of decompression to the classical addition cost, so the cost of this new addition is requires 3I+2M+4S.

Table 6: Decompression, Addition cost in Montgomery curve

coordinate	Classical addition	Decompression	New Addition
y	I+2M+S	I+2M+2S	2I+4M+3S
x with a=0	I+2M+S	2I+0M+3S	3I+2M+4S
x or y with $a \neq 0$	I+2M+S	2I+0M+3S	3I+2M+4S

In the table above we see that the cost of addition of two points in Montgomery curve with y-compression change from I+2M+S to 2I+4M+3S to save 25% of memory.
- Also, with x-coordinate compression, the cost of addition of two points in Montgomery curve change from I+2M+S to 3I+2M+4S to save 25% of memory.

GLOBAL COMPARISON:

In the table below we see that the cost of new addition with $(x_1, x_2, y_1 + y_2)$ compression is better than $(x_1 + x_2, y_1, y_2)$ compression.
The complexity of addition:
In [3], to compute multiplication we have a complexity of $O(mlogm)$ and to compute inversion we have a compexity of $O(mlog^2mloglogm)$. Our table in term of complicity become

Table 7: The cost of addition

Coordinate	Classical addition	$(x_1, x_2, y_1 + y_2)$	$(x_1 + x_2, y_1, y_2)$
Affine	I+2M+S	2I+4M+3S	3I+2M+4S
Projective	12M+2S	3I+16M+6S	3I+22M+7S
Jacobian	12M+4S	I+18M+12S	3I+24M+13S
Edwards curve	2I+5M	3I+9M+2S	3I+9M+2S
Montgomery curve	I+2M+S	2I+4M+3S	3I+2M+4S

Table 8: The complexity of addition

Addition	No compression	$(x_1, x_2, y_1 + y_2)$ compression	$(x_1 + x_2, y_1, y_2)$ compression
Affine	O(mlogm(3+logmloglogm))	O(mlogm(7+2logmloglogm))	O(mlogm(10+3logmloglogm))
Projective	O(14mlogm)	O(mlogm(22+3logmloglogm))	O(mlogm(28+3logmloglogm))
Jacobian	O(18mlogm)	O(mlogm(30+logmloglogm))	O(mlogm(37+3logmloglogm))
Edwards curve	O(mlogm(5+2logmloglogm))	O(mlogm(11+3logmloglogm))	O(mlogm(11+3logmloglogm))
Montgomery curve	O(mlogm(3+logmloglogm))	O(mlogm(7+2logmloglogm))	O(mlogm(6+3logmloglogm))

4 CONCLUSION

In this work, we provided details and important improvements of two new methods of compression points in a field of characteristic $\neq 2, 3$ with affine, Jacobian and projective coordinate, also provide the cost of those operations. We see that if our method costs the same circuit delay, our method can achieve better implementation efficiency compared with the classical method, so it will save 25% of memory size if we work with affine coordinate and at least 17% if we work with Jacobian or Projective coordinate. Also we see that we work with this compression point $(x_1, x_2, \alpha = y_1 + y_2)$ is a little bit better than the other compression point $(\alpha = x_1 + x_2, y_1, y_2)$.

REFERENCES

[1] N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation 48, pp. 203-209, 1987.

[2] V.S. Miller. Use of Elliptic Curves in Cryptography. In: In Advances in Cryptology - Crypto'85. volume 218 of LNCS, pp. 417-426, Springer-Verlag, 1986.

[3] S. B. Gashkov and I. S. Sergeev. COMPLEXITY OF COMPUTATION IN FINITE FIELDS. Journal of Mathematical Sciences, Vol. 191, No. 5, June, 2013.

[4] PAULO S. L. M. BARRETO and JOSE FELIPE VOLOCH. Efficient Computation of Roots in Finite Fields. 2004 Kluwer Academic Publishers.

[5] M. Khabbazian, T. Gulliver, and V. Bhargava, "Double point compression with applications to speeding up random point multiplication", IEEE Transactions on Computers, vol. 56, no. 3, pp. 305-313, 2007.

[6] S.B.Gashkov, I.B.Gashkov. Fast algorithm of square rooting in some finite field of odd characteristic. ISSN 0027-1322, Moscow university Mathematics Bulletin, 2018, vol. 73. No 5, pp. 176-181.

[7] Majid Khabbazian, Student Member, IEEE, T. Aaron Gulliver, Senior Member, IEEE, and Vijay K. Bhargava, Fellow, IEEE. Double Point Compression with Applications to Speeding Up Random Point Multiplication. IEEE TRANSACTIONS ON COMPUTERS, VOL. 56, NO. 3, MARCH 2007.

[8] Alina Dudeanu, George-Razvan Oancea, Sorin Iftene. An x-Coordinate Point Compression Method for Elliptic Curves over Fp. 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing.

[9] Xinxin Fan, Adilet Otemissov, Francesco Sica, Andrey Sidorenko. Multiple point compression on elliptic curves. Codes Cryptogr. (2017) 83:565-588.

[10] Steven D. Galbraith, Xibin Lin. Computing pairings using x-coordinates only. Codes Cryptogr. (2009) 50:305-324.

[11] Billy Bob Brumley, Kimmo U. Jarvinen. Fast Point Decompression for Standard Elliptic Curves. S.F. Mjolsnes, S. Mauw, and S.K. Katsikas (Eds.): EuroPKI 2008, LNCS 5057, pp. 134âĂŞ149, 2008. Springer-Verlag Berlin Heidelberg 2008.

[12] YU ZHANG, YIN LI, QING CHEN. Fast Asymptotic Square Root for Two Types of Special Pentanomials. Digital Object Identifie 10.1109/ACCESS.2019.2911012.

[13] Nadia El Mrabet, Marc Joye. Guide to pairing-based cryptography. Chapman and Hall/CRC Cryptography and network security. (2017)

[14] H.M. Edwards. A normal form for elliptic curves. Bulletin of the American Mathematical Society, 44:393-422, 2007.

[15] Benjamin Justus. Point Compression and Coordinate Recovery for Edwards Curves over Finite Field. Analele University de Vest, Timisoara Seria Matematica Informatica LII, 2, (2014), 111-125.

[16] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Compression Point in Field of Characteristic 3. Springer, I4CS 2022, CCIS 1747, pp. 104âĂŞ111, 2022 https://doi.org/10.1007/978-3-031-23201-5_7.

[17] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Tower Building Technique on Elliptic Curve with Embedding Degree 36. WSEAS TRANSACTIONS ON COMPUTERS. DOI: 10.37394/23205.2022.21.39.

[18] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Tower Building Technique on Elliptic Curve with Embedding Degree 72. WSEAS Transactions on Computer Research 10:126-138 DOI: 10.37394/232018.2022.10.17

[19] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. TOWER BUILDING TECHNIQUEON ELLIPTIC CURVEWITH EMBEDDING DEGREE 18. Tatra mountains mathematical publications, DOI: 10.2478/tmmp-2023-0008Tatra Mt. Math. Publ. 83 (2023), 103âĂŞ118.

[20] ISMAIL ASSOUJAA, SIHAM EZZOUAK, HAKIMA MOUANIS. Pairing based cryptography New random point exchange key protocol. Conference: 2022 7th International Conference on Mathematics and Computers in Sciences and Industry (MCSI), DOI: 10.1109/MCSI55933.2022.00017.