

# Elliptic Curve Cryptograph

## - Learning ECC -

Ji Yong-Hyeon

**Department of Information Security, Cryptology, and Mathematics**

College of Science and Technology  
Kookmin University

February 17, 2024



## Acknowledgements

# Contents

- 1 NIST P256 . . . . . 1**
  - 1.1 Data Representation . . . . . 2
  - 1.2 secp256r1 . . . . . 3
  - 1.3 Multi-precision Addition . . . . . 6
    - 1.3.1 Theory for Addition . . . . . 6
    - 1.3.2 Practice for Addition . . . . . 8
- 2 Elliptic Curve Theory . . . . . 9**
  - 2.1 A Puzzle of Squares and Pyramids . . . . . 9
    - 2.1.1 Diophantus' Approach . . . . . 10
    - 2.1.2 Extending Diophantus' Method . . . . . 10
  - 2.2 Why is it called an Elliptic Curve? . . . . . 11
    - 2.2.1 Abel's Insight . . . . . 11
    - 2.2.2 The Geometry of an Ellipse . . . . . 11
    - 2.2.3 The Arc Length of an Ellipse . . . . . 11
    - 2.2.4 Connecting to an Elliptic Curve . . . . . 11
- 3 Elliptic Curves in Cryptography . . . . . 18**
- A Additional Data A . . . . . 26**
  - A.1 Existence of an Additional Root in Cubic Functions via the Intermediate Value Theorem 26

# Chapter 1

## NIST P256

### Configuration

```
1  #ifdef _WIN32 // Windows-specific definitions
2  #include <windows.h>
3  #include <stdint.h>
4  /* ... */
5  typedef DWORD      u32;
6  typedef DWORDLONG  u64;
7  #elif defined(__linux__) // Linux-specific definitions
8  #include <stdint.h>
9  typedef int8_t      i8;
10 typedef int32_t      i32;
11 typedef int64_t      i64;
12 typedef uint8_t      u8;
13 typedef uint32_t     u32;
14 typedef uint64_t     u64;
15 #else
16 #error "Unsupported platform"
17 #endif
18
19 #if !defined(_WIN64) && !defined(__x86_64__)
20 #define IS_32_BIT_ENV
21 #endif
22
23 #ifdef IS_32_BIT_ENV // 32-bit specific settings
24 #define ONE          0x01
25 #define SIZE          8
26 typedef u32          word;
27 typedef u32          field[SIZE];
28 #else // 64-bit specific settings
29 #define ONE          0x01LL
30 #define SIZE          4
31 typedef u64          word;
32 typedef u64          field[SIZE];
33 #endif
```

## 1.1 Data Representation

128-bit Hexa-string	0x77FDDC58464B01FC6606BC465BF5CBCB											
String Index	0	...	7	8	...	15	16	...	23	24	...	31
Split into Words	77FDDC58 data[3]			464B01FC data[2]			6606BC46 data[1]			5BF5CBCB data[0]		
data[0]	(data[0] = '5'-'0';) -> (data[0] <= 4;) -> (data[0]  = 'B'-'0';) -> (data[0] <= 4;) -> (data[0]  = 'F'-'0';) -> ...											
	0x5 -> 0x50 -> 0x5B -> 0x5B0 -> ...											

```

1 void stringToWord(word* wordArray, const char* hexString) {
2     size_t length = strlen(hexString) / (SIZE == 8 ? 8 : 16);
3     if (length != SIZE) {
4         printf("Invalid 128-bit Hexa-string Length!\n");
5         return;
6     }
7     for (size_t i = 0; i < length; i++)
8 #ifdef IS_32_BIT_ENV
9         sscanf(&hexString[i * 8], "%8X",
10             &wordArray[(length - 1) - i]);
11 #else
12         sscanf(&hexString[i * 16], "%16lX",
13             &wordArray[(length - 1) - i]);
14 #endif
15 }

```

```

1 int main(void) {
2     const char* string = "
3         BD91C935C85617B079C6F2728B987CE4
4         88BB17B4644D5F8B9C23AF955AB74663
5     ";
6
7     word data[SIZE];
8     stringToWord(data, string);
9
10    for (i32 i = SIZE-1; i >=0; i--)
11 #ifdef IS_32_BIT_ENV
12        printf("%8X:", data[i]);
13 #else
14        printf("%16lX:", data[i]);
15 #endif
16    puts("");
17 }

```

```

BD91C935:C85617B0:79C6F272:8B987CE4:88BB17B4:644D5F8B:9C23AF95:5AB74663:
BD91C935C85617B0:79C6F2728B987CE4:88BB17B4644D5F8B:9C23AF955AB74663:

```

**Example 1.1** (secp256r1). Consider  $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ .

$2^{256}$	00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
$-2^{224}$		00000001	00000000	00000000	00000000	00000000	00000000	00000000	00000000
$+2^{192}$			00000001	00000000	00000000	00000000	00000000	00000000	00000000
$+2^{96}$						00000001	00000000	00000000	00000000
$-2^0$									00000001
$p$	FFFFFFFF	00000001	00000000	00000000	00000000	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
$p_{\text{inv}}$	00000000	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	00000000	00000000	00000000	00000001

```

1  #ifdef IS_32_BIT_ENV
2  static const field PRIME = {
3      0xFFFFFFFFU, 0xFFFFFFFFU, 0xFFFFFFFFU, 0x00000000U,
4      0x00000000U, 0x00000000U, 0x00000001U, 0xFFFFFFFFU
5  };
6  static const field PRIME_INVERSE = {
7      0x00000001U, 0x00000000U, 0x00000000U, 0xFFFFFFFFU,
8      0xFFFFFFFFU, 0xFFFFFFFFU, 0xFFFFFFFFU, 0x00000000U
9  };
10 #else
11 static const field PRIME = {
12     0xFFFFFFFFFFFFFFFFFU, 0x00000000FFFFFFFFFU,
13     0x0000000000000000U, 0xFFFFFFFF00000001U
14 };
15 static const field PRIME_INVERSE = {
16     0x0000000000000000U, 0xFFFFFFFF00000000U,
17     0xFFFFFFFFFFFFFFFFFU, 0x00000000FFFFFFFFFU
18 };
19 #endif

```

## 1.2 secp256r1

Define a prime number

$$\begin{aligned}
 p_{256} &= 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1 \\
 &= 2^{32 \cdot 8} - 2^{32 \cdot 7} + 2^{32 \cdot 6} + 2^{32 \cdot 3} - 2^{32 \cdot 0}
 \end{aligned}$$

that is used in the context of cryptography, particularly in the construction of elliptic curves for cryptographic purposes. For prime  $p$ , let

$$m = \lceil \log_2 p \rceil, \quad t = \left\lceil \frac{m}{W} \right\rceil.$$

For example, for  $p = 2^{256}$ , we have  $t = \left\lceil \frac{\lceil \log_2 2^{256} \rceil}{2^{32}} \right\rceil$ . Note that

- $A = A[t-1] \parallel \cdots \parallel A[2] \parallel A[1] \parallel A[0]$
- $a = 2^{(t-1)W} A[t-1] + \cdots + 2^{2W} A[2] + 2^W A[1] + A[0]$

sign	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	Decimal	One's Complement				Two's Complement			
0	0	0	0	+0	0	0	0	0	0	0	0	0
0	0	0	1	+1	0	0	0	1	0	0	0	1
0	0	1	0	+2	0	0	1	0	0	0	1	0
0	0	1	1	+3	0	0	1	1	0	0	1	1
0	1	0	0	+4	0	1	0	0	0	1	0	0
0	1	0	1	+5	0	1	0	1	0	1	0	1
0	1	1	0	+6	0	1	1	0	0	1	1	0
0	1	1	1	+7	0	1	1	1	0	1	1	1
1	0	0	0	-0	1	1	1	1	0	0	0	0
1	0	0	1	-1	1	1	1	0	1	1	1	1
1	0	1	0	-2	1	1	0	1	1	1	1	0
1	0	1	1	-3	1	1	0	0	1	1	0	1
1	1	0	0	-4	1	0	1	1	1	1	0	0
1	1	0	1	-5	1	0	1	0	1	0	1	1
1	1	1	0	-6	1	0	0	1	1	0	1	0
1	1	1	1	-7	1	0	0	0	1	0	0	1
				-8					1	0	0	0

```

1  #ifdef _64BIT_SYSTEM
2  typedef u64 field_element[4]; // For 64-bit systems
3  #else
4  typedef u32 field_element[8]; // For 32-bit systems
5  #endif
6
7  // Example for modular addition (simplified)
8  void mod_add(field_element a, field_element b, field_element
   result) {
9      uint64_t carry = 0;
10     for (int i = 0; i < 4; ++i) { // Assuming 64-bit system
11         uint64_t temp = (uint64_t)a[i] + b[i] + carry;
12         result[i] = temp & 0xFFFFFFFFFFFFFFFF; // Keep only 64
           bits
13         carry = temp >> 64; // Carry for the next iteration
14     }
15
16     // Modular reduction if necessary
17     if (carry || is_greater_or_equal(result, p256)) {
18         // Subtract p256 if result >= p256
19         subtract_p256(result);
20     }
21 }
22

```



```
23 void subtract_p256(field_element x) {  
24     // This is a simplified version. In practice, you'd need to  
    handle underflows.  
25     // Subtract ( $2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ )  
26     // In practice, implement this function based on the specific  
    structure of p256  
27     // and considering the binary representation of the field  
    elements.  
28 }
```

## 1.3 Multi-precision Addition

### 1.3.1 Theory for Addition

**Note.** A positive integer  $X \in [2^{w(n-1)}, 2^{wn})$  is a  $n$ -word string. For example, let  $w = 32$  and consider 4-word string

$$\begin{array}{|c|c|c|c|} \hline x[3] \cdot 2^{w \cdot 3} & x[2] \cdot 2^{w \cdot 2} & x[1] \cdot 2^{w \cdot 1} & x[0] \cdot 2^{w \cdot 0} \\ \hline \end{array}$$

Then

$$\begin{array}{l} \text{Minimum} \\ \text{Maximum} \end{array} \begin{array}{|c|c|c|c|} \hline 00000001 & 00000000 & 00000000 & 00000000 \\ \hline \text{FFFFFFFF} & \text{FFFFFFFF} & \text{FFFFFFFF} & \text{FFFFFFFF} \\ \hline \end{array} \begin{array}{l} = 2^{w \cdot 3} \\ = 2^{w \cdot 4} - 1 \end{array}$$

#### Upper and Lower Bound of Addition

**Proposition 1.1.** Let  $w \in \{32, 64\}$  be a word. Let  $X$  and  $Y$  are  $n$ -word and  $m$ -word strings, respectively, i.e.,

$$\begin{aligned} X &= x[n-1] \parallel \cdots \parallel x[0] \in [2^{w(n-1)}, 2^{wn}) \quad \text{with } x[n-1] \neq 0 \\ Y &= y[m-1] \parallel \cdots \parallel y[0] \in [2^{w(m-1)}, 2^{mn}) \quad \text{with } y[m-1] \neq 0 \end{aligned}$$

Then

$$2^{w \cdot (\max(n,m)-1)} < X + Y < 2^{w \cdot (\max(m,n)+1)}.$$

*Proof.* Let  $W := 2^w$ . Then  $X$  and  $Y$  can be expressed as follows:  $\begin{cases} X = xW^{n-1} + X' \\ Y = yW^{m-1} + Y' \end{cases}$  where

$$a, b \in (0, W), \quad A' \in [0, W^{n-1} - 1], \quad B' \in [0, W^{m-1} - 1].$$

Suppose that  $n \geq m$  then

$$\begin{aligned} W^{n-1} &\leq \max(A, B) < A + B = (aW^{n-1} + A') + (bW^{m-1} + B') \\ &< (a + b)W^{n-1} + (W^{n-1} + W^{n-1}) \\ &= (a + b + 2)W^{n-1} \\ &\leq ((W - 1) + (W - 1) + 2)W^{n-1} \\ &= 2W^n \leq W^{n+1}. \end{aligned}$$

Thus  $W^{n-1} < A + B < W^{n+1}$ . Here,  $n = \max(n, m)$ . □

#### Corollary 1.1.1.

$$\text{len}_{\text{word}}(X) = t = \text{len}_{\text{word}}(Y) \implies \text{len}_{\text{word}}(X + Y) \leq t + 1.$$

**Single-Word Addition  $x[i] + y[i]$** **Proposition 1.2.** *Let  $x, y \in [0, 2^w)$  are single-words.*

(1)  $\text{len}_{\text{word}}(x + y) \leq 2.$

(2) (Division Theorem)

$$x + y = q2^w + r = \left\lfloor \frac{x + y}{2^w} \right\rfloor \cdot 2^w + (x \boxplus y).$$

(3)  $(\text{carry}) \left\lfloor \frac{x+y}{2^w} \right\rfloor \in \{0, 1\}.$

(4)  $(\star) x \boxplus y < x \Leftrightarrow \left\lfloor \frac{x+y}{2^w} \right\rfloor = 1.$

**Note.**

$x \boxplus y < x$	True	carry = 1
	False	carry = 0

**Single-Word Addition with Carry  $x[i] \boxplus y[i] + \varepsilon$** **Proposition 1.3.** *Let  $x, y \in [0, 2^w)$  and  $\varepsilon \in \{0, 1\}.$* 

(1)  $\text{len}_{\text{word}}((x \boxplus y) + \varepsilon) \leq 2.$

(2)  $(\text{carry}) \left\lfloor \frac{(x \boxplus y) + \varepsilon}{2^w} \right\rfloor \in \{0, 1\}.$

(3)
$$\left\lfloor \frac{x + y}{2^w} \right\rfloor = 1 \implies \left\lfloor \frac{(x \boxplus y) + \varepsilon}{2^w} \right\rfloor = 0.$$

**Note.**

$x \boxplus y < x$	True	carry = 1	$\implies$
	False	carry = 0	

### 1.3.2 Practice for Addition

---

**Algorithm 1:** Multi-Precision Addition
 

---

**Input:**  $X, Y \in [0, 2^{wt}) \subseteq \mathbb{Z}$  //  $(w, t) = (32, 8)$  or  $(w, t) = (64, 4)$  for secp256r1

**Output:**  $(\varepsilon, Z)$  where  $Z = (X + Y) \bmod 2^{wt}$  and  $\varepsilon \in \{0, 1\}$  is carry bit

```

1 Function ADDITION_CORE( $\varepsilon, Z, X, Y$ ):
2    $(\varepsilon, z[0]) \leftarrow x[0] + y[0]$ 
3   for  $i = 1$  to  $t - 1$  do
4      $(\varepsilon, z[i]) \leftarrow x[i] + y[i] + \varepsilon$ 
5   end
6   return  $(\varepsilon, Z = z[t - 1] \parallel z[t - 2] \parallel \dots \parallel z[0])$ 
7 end

```

---

**Example 1.2.**

X		0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
Y	+	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF
$\varepsilon$	1	1	1	1	0
Z		0x00000001	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

**Note** (How to Compute Carry?). content...

---

**Algorithm 2:** Addition in  $\mathbb{F}_p$ 


---

**Input:** Modulus  $p$ , and integer  $X, Y \in [0, p)$

**Output:**  $Z = (X + Y) \bmod p$

```

1  $(\varepsilon, Z) \leftarrow x[0] + y[0]$ 
2 for  $i = 1$  to  $t - 1$  do
3    $(\varepsilon, z[i]) \leftarrow x[i] + y[i] + \varepsilon$ 
4 end
5 return  $(\varepsilon, Z = z[t - 1] \parallel z[t - 2] \parallel \dots \parallel z[0])$ 

```

---

**Example 1.3.**

$\varepsilon$	1	1	1	1	1	1	1	0	0
X		BD91C935	C85617B0	79C6F272	8B987CE4	88BB17B4	644D5F8B	9C23AF95	5AB74663
Y	+	4E272A73	41569559	F3E58053	BE961728	D67BF71E	FBA44BF2	83DAA7ED	9BF6DDA8
Z	1	0BB8F3A9	09ACAD0A	6DAC72C6	4A2E940D	5F370ED3	5FF1AB7E	1FFE5782	F6AE240B
$-p$		FFFFFFFF	00000001	00000000	00000000	00000000	FFFFFFFF	FFFFFFFF	FFFFFFFF
$Z - p$		0BB8F3AA	09ACAD09	6DAC72C6	4A2E940D	5F370ED2	5FF1AB7E	1FFE5782	F6AE240C

Note that

$\varepsilon$	0	1	1	1	1	0	0	0	0
Z	1	0BB8F3A9	09ACAD0A	6DAC72C6	4A2E940D	5F370ED3	5FF1AB7E	1FFE5782	F6AE240B
$+p_{\text{inv}}$		00000000	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF	00000000	00000000	00000001
$Z + p_{\text{inv}}$	1	0BB8F3AA	09ACAD09	6DAC72C6	4A2E940D	5F370ED2	5FF1AB7E	1FFE5782	F6AE240C

# Chapter 2

## Elliptic Curve Theory

### 2.1 A Puzzle of Squares and Pyramids

Consider the following question:

“What is the number of balls that may be piled as a square pyramid and also re-arranged into a square array?”

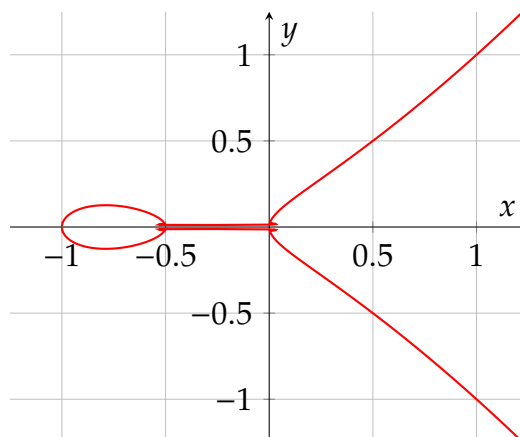
To address this, let  $x$  be the height of the pyramid. The number of balls in a pyramid of height  $x$  is given by:

$$1^2 + 2^2 + 3^2 + \dots + x^2 = \frac{x(x+1)(2x+1)}{6}$$

We seek a configuration where this sum also forms a perfect square, i.e.,

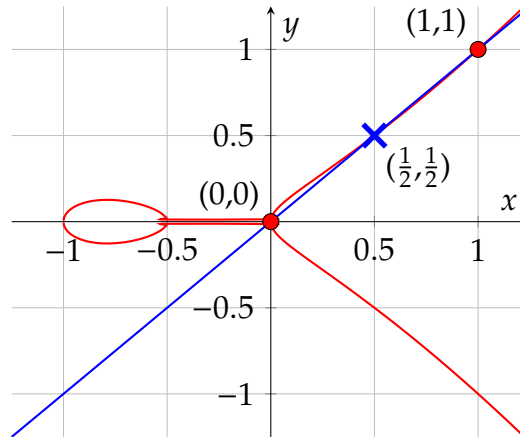
$$y^2 = \frac{x(x+1)(2x+1)}{6}$$

This equation forms the basis of our puzzle, intertwining the concepts of geometric and numeric squares.



### 2.1.1 Diophantus' Approach

We consider a set of known points to produce new points. The trivial solutions  $(0, 0)$  and  $(1, 1)$  fit the equation of the line  $y = x$ .



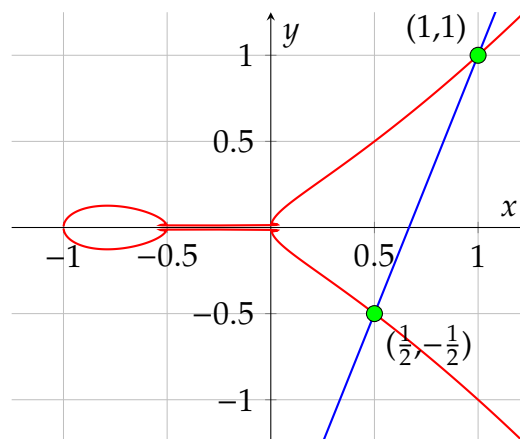
Intersecting this line with the curve described by our pyramid problem, we rearrange terms:

$$\begin{aligned}\frac{x(x+1)(2x+1)}{6} &= x^2, \\ (x^2+x)(2x+1) &= 6x^2, \\ 2x^3+x^2+2x^2+x &= 6x^2, \\ x(2x^2-3x+1) &= 0, \\ x(x-1)(2x-1) &= 0.\end{aligned}$$

We find that  $x = \frac{1}{2}$  is a solution, implying  $y = \frac{1}{2}$ . The symmetry of the curve also yields  $(\frac{1}{2}, -\frac{1}{2})$  as another solution.

### 2.1.2 Extending Diophantus' Method

Consider the line through  $(\frac{1}{2}, -\frac{1}{2})$  and  $(1, 1)$ , which implies  $y = 3x - 2$ .



Intersecting this with our curve, we derive:

$$x^3 - \frac{51}{2}x^2 + \dots = 0 \tag{2.1}$$

This leads to the solutions  $x = 24$  and  $y = 70$ , demonstrating the power of algebraic manipulation and geometric insight.

## 2.2 Why is it called an Elliptic Curve?

The term “elliptic curve” has its roots in the quest to measure the circumference of an ellipse. Consider the trigonometric function  $y = \sin w$ . The inverse function,  $w(y) = \sin^{-1} y$ , is expressed as an integral:

$$w(y) = \sin^{-1} y = \int_0^y \frac{1}{\sqrt{1-t^2}} dt$$

This integral is foundational in understanding the link between elliptic curves and elliptic integrals.

### 2.2.1 Abel’s Insight

Niels Henrik Abel, a prominent mathematician, extended this concept. Starting with  $y = \sin w$ , Abel explored the inverse functions of elliptic integrals, uncovering their double periodicity. He defined the function:

$$F(w) = \int_0^w \frac{dz}{\sqrt{(1-z^2)(1-k^2z^2)}}$$

Abel’s work laid the groundwork for understanding the complex nature of elliptic curves.

### 2.2.2 The Geometry of an Ellipse

An ellipse is defined by the equation  $x^2/a^2 + y^2/b^2 = 1$ . This simple equation belies the complexity of calculating its arc length.

### 2.2.3 The Arc Length of an Ellipse

Defining  $k^2 = 1 - \frac{b^2}{a^2}$  and changing variables  $x \rightarrow ax$ , we express the arc length of an ellipse as:

$$a \int_{-1}^1 \sqrt{\frac{1-k^2x^2}{1-x^2}} dx = a \int_{-1}^1 \frac{1-k^2x^2}{\sqrt{(1-x^2)(1-k^2x^2)}} dx$$

This leads to the following representation of the arc length:

$$\text{Arc Length} = a \int_{-1}^1 \frac{1-k^2x^2}{y} dx \quad \text{with} \quad y^2 = (1-x^2)(1-k^2x^2).$$

### 2.2.4 Connecting to an Elliptic Curve

The ellipse’s arc length calculation brings us to a critical realization. An elliptic integral is generally expressed as:

$$\int R(x, y) dx$$

This integral, deeply connected to the geometry of ellipses, underpins the theory of elliptic curves.

## Double Periodicity in Elliptic Curves

Elliptic curves are intimately connected with the study of complex tori, which can be represented through the use of doubly periodic functions. A fundamental example of such a function is the Weierstrass  $\wp$  function, defined by a lattice  $\Lambda$  in the complex plane.

### The Weierstrass $\wp$ Function

Given a lattice  $\Lambda \subset \mathbb{C}$ , the Weierstrass  $\wp$  function is defined as:

$$\wp(z; \Lambda) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left( \frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right). \quad (2.2)$$

This function is even,  $\wp(-z) = \wp(z)$ , and exhibits double periodicity with respect to the lattice  $\Lambda$ , meaning:

$$\wp(z + \omega) = \wp(z) \quad \text{for all } \omega \in \Lambda. \quad (2.3)$$

### Elliptic Curves and the $\wp$ Function

An elliptic curve can be associated with the Weierstrass  $\wp$  function. Specifically, an elliptic curve over  $\mathbb{C}$  can be described in the Weierstrass form:

$$y^2 = 4x^3 - g_2x - g_3, \quad (2.4)$$

where  $g_2$  and  $g_3$  are constants derived from the lattice  $\Lambda$ . The coordinates  $(x, y)$  on the elliptic curve correspond to the values of the Weierstrass  $\wp$  function and its derivative:

$$x = \wp(z; \Lambda), \quad y = \wp'(z; \Lambda). \quad (2.5)$$

## Double Periodicity

### Two linearly independent periods.

$$\phi(z + w_1) = \phi(z + w_2) = \phi(z) \quad \text{for all complex number } z.$$

### It satisfies

$$[\phi'(z)]^2 = 4\phi(z)^3 - 60G_4\phi(z) - 140G_6$$

- So for  $x = \phi(z)$  and  $y = \phi'(z)$
- $y^2 = 4x^3 - 60G_3x - 140G_6$



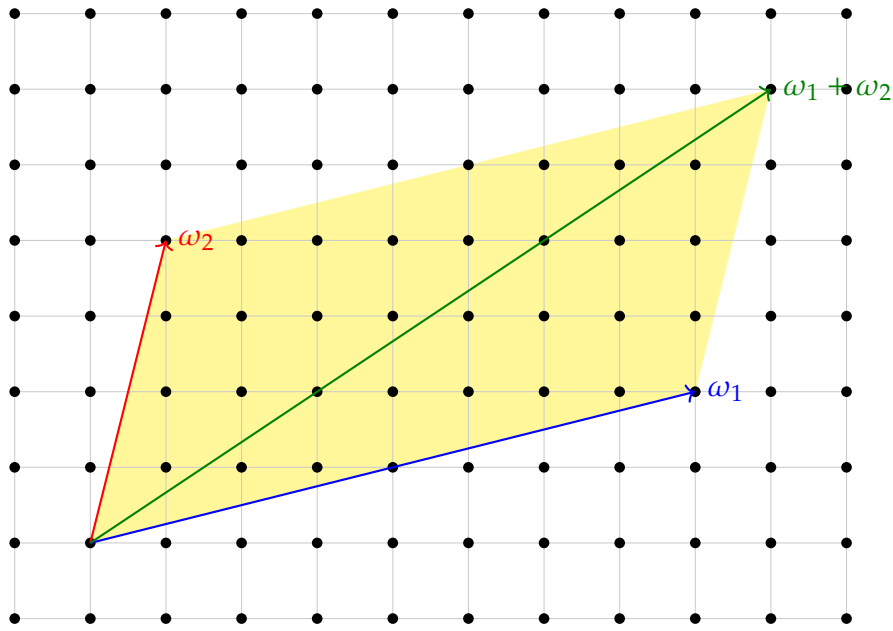
## Elliptic Functions and Elliptic Curves

The  $\wp$ -function and its derivative satisfy an algebraic relation

$$\wp'(z)^2 = \wp(z)^3 + A\wp(z) + B$$

The double periodicity means that it is a function on the quotient space  $\mathbb{C}/\Lambda$ , where  $\Lambda$  is the lattice

$$\Lambda = \{n_1\omega_1 + n_2\omega_2 : n_1, n_2 \in \mathbb{Z}\}.$$



The lattice  $L$  is generated by  $\omega_1$  and  $\omega_2$  in the quotient space  $\mathbb{C}/L$ .

## Elliptic Functions and Elliptic Curves

Elliptic functions and elliptic curves are fundamental objects in complex analysis and algebraic geometry, respectively. They are interconnected through the Weierstrass  $\wp$  function and its properties.

### Weierstrass Elliptic Functions

The Weierstrass elliptic functions are defined with respect to a lattice  $\Lambda \subset \mathbb{C}$ . The Weierstrass  $\wp$  function, a key example of an elliptic function, is defined as:

$$\wp(z; \Lambda) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left( \frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right). \quad (2.6)$$

This function is doubly periodic and meromorphic with poles of order two at lattice points.

## Elliptic Curves and the Weierstrass $\wp$ Function

An elliptic curve can be described as a set of points satisfying a cubic equation in two variables. Over the complex numbers, this curve can be associated with the Weierstrass  $\wp$  function.

An elliptic curve in Weierstrass form is given by:

$$y^2 = 4x^3 - g_2x - g_3, \quad (2.7)$$

where  $g_2$  and  $g_3$  are constants determined by the lattice  $\Lambda$ . The function  $\wp$  and its derivative relate to the curve as follows:

$$x = \wp(z; \Lambda), \quad (2.8)$$

$$y = \wp'(z; \Lambda). \quad (2.9)$$

This establishes a correspondence between points on the complex torus  $\mathbb{C}/\Lambda$  and points on the elliptic curve.

## Properties of Elliptic Curves

Elliptic curves have several important properties:

- They form a group under a geometrically defined addition operation.
- The addition operation on the curve corresponds to the addition of points in the complex plane modulo the lattice  $\Lambda$ .
- Elliptic curves over finite fields have applications in number theory and cryptography.

## The Complex Points on an Elliptic Curve

The  $\phi$ -function gives a complex analytic isomorphism

$$\frac{\mathbb{C}}{L} = (\phi(z), \phi'(z)) \rightarrow E(\mathbb{C})$$

with the notation that  $\mathbb{C}$  is the complex numbers,  $L$  is a lattice, and  $E(\mathbb{C})$  is the set of complex points on an elliptic curve.

Thus the points of  $E$  with coordinates in the complex numbers  $\mathbb{C}$  form a *torus*, that is, the surface of a donut.

## $X^2 + Y^2 = C$

- Let  $x = a + b\sqrt{-1}$ ,  $y = c + d\sqrt{-1}$ .
- The solution over complex numbers is a surface, in fact topologically sphere.
- If unbelievable, check out level curves.
- Furthermore, it has group structure.

$$(a + b\sqrt{-1})(c + d\sqrt{-1}) \text{ becomes } ac - bd + (ad + bc)\sqrt{-1}$$

## Why is it called Torus?

- Complex Tori

$$y^2 = x(x^2 - 1)$$

- If we introduce *points at infinity* and the *complex numbers*, we can argue that the graph is a torus.

## Why Elliptic Curve?

- Discrete Logarithm Problem
- Given a finite group  $G$  with two of its elements  $a$  and  $b$ .
- Find an integer  $x$  such that,  $a^x = b$  if it exists.
- Example: Non-zero elements of some finite field.

## Better groups?

For a finite field  $F$ ,

$$GL_2(F) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid ad - bc \neq 0, a, b, c, d \in F \right\}$$

- The Times(London) Jan. 1999 An Irish schoolgirl Sarah Flannery used matrices as an alternative to RSA. Her algorithm is far faster than the RSA and equally secure.

- The Art of Computer Programming by Donald Knuth

How about this group?

- $F = \mathbb{Z}/17\mathbb{Z} = \mathbb{Z} \pmod{17}$
- $6^2 = 36 \equiv 2 \pmod{17}$
- 6 behaves like  $\sqrt{2}$

$$X^2 - 2Y^2 = 1$$

$$(3 + 2\sqrt{2})(3 - 2\sqrt{2}) = 1$$

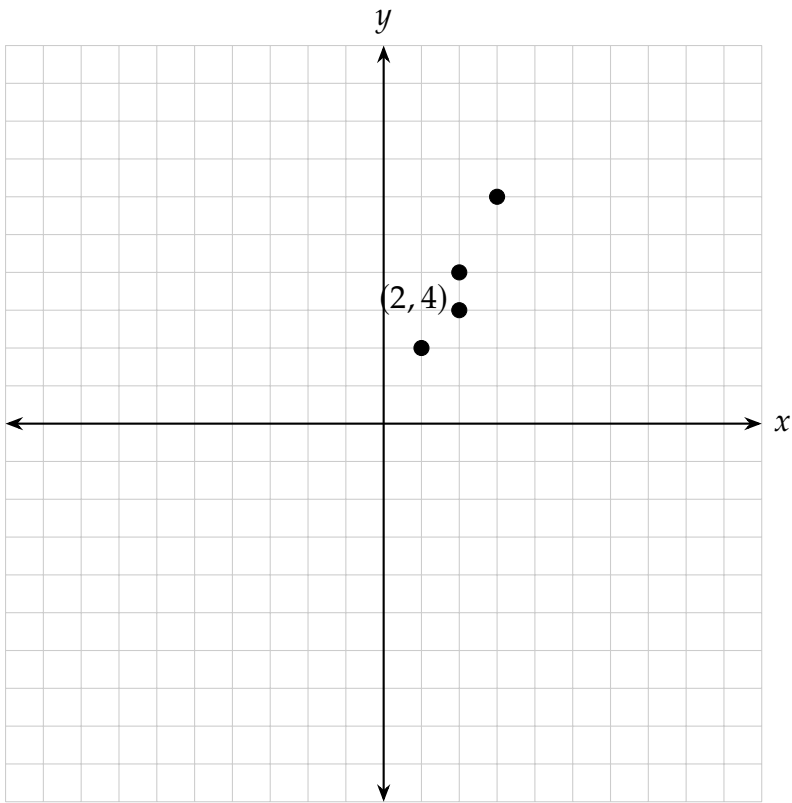
$$(3 + 12)(3 - 12) = -36 \equiv 1 \pmod{17}$$

- Let  $G = \{(x, y) \mid x^2 - 2y^2 = 1 \text{ over } \mathbb{F}\}$  The operation on  $G$  is defined as:

$$\begin{aligned} (x_1, y_1) \cdot (x_2, y_2) &= \\ (x_1 + \sqrt{2}y_1)(x_2 + \sqrt{2}y_2) &= \\ = (x_1x_2 + 2y_1y_2) + \sqrt{2}(x_1y_2 + x_2y_1) &= \\ (x_1, y_1) \cdot (x_2, y_2) &= \\ = (x_1x_2 + 2y_1y_2, x_1y_2 + x_2y_1) \end{aligned}$$

## Why Elliptic Curve?

- DLP (Discrete Logarithm Problem) on finite field can be solved faster than we thought!
- by “index calculus”
- To protect against this attack...
- Elliptic curves!



## Chapter 3

# Elliptic Curves in Cryptography

- Elliptic Curve (EC) cryptography were first proposed in 1985 independently by Neal Koblitz and Victor Miller.
- The **discrete logarithm** problem on elliptic curve groups is believed to be more difficult than the corresponding problem in the multiplicative group of non-zero elements of the underlying finite field.

### On finite fields

Consider  $y^2 \equiv x^3 + 2x + 3 \pmod{5}$

$x = 0, y^2 = 3$	no solution (mod 5)
$x = 1, y^2 = 6 \equiv 1,$	$y = 1, 4 \pmod{5}$
$x = 2, y^2 = 15 \equiv 0,$	$y = 0 \pmod{5}$
$x = 3, y^2 = 36 \equiv 1,$	$y = 1, 4 \pmod{5}$
$x = 4, y^2 = 75 \equiv 0,$	$y = 0 \pmod{5}$

Then points on the elliptic curve are  $(1, 1), (1, 4), (2, 0), (3, 1), (3, 4), (4, 0)$  and the point at infinity. Denote it by  $O$ .

### Notation

- $GF(q)$  or  $\mathbb{F}_q$ : finite field with  $q$  elements, typically,  $q = p$  where  $p$  is prime, or  $2^m$ .
- $E(\mathbb{F}_q)$ : elliptic curve over  $\mathbb{F}_q$ .
- $(x, y)$ : point on  $E(\mathbb{F}_q)$ .
- $O$ : point at infinity.

## Definition of Elliptic curves

- An elliptic curve over a field  $K$  is a non-singular cubic curve in two variables,  $f(x, y) = 0$  with a rational point (which may be a point at infinity).
- The field  $K$  is usually taken to be the complex numbers, reals, rationals, algebraic extensions of rationals,  $p$ -adic numbers, or a *finite field*.
- Elliptic curves groups for cryptography are examined with the underlying fields of  $\mathbb{F}_p$  (where  $p > 3$  is a prime) and  $\mathbb{F}_{2^m}$  (a binary representation with  $2^m$  elements).

## EC

An *elliptic curve* is a plane curve defined by an equation of the form, when characteristic is neither 2 nor 3, and ... What the hell?

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

## Hmm...

- $x^3 + y^3 + 1 = 0$  is a cubic curve...?
- Let  $x = u + v$ ,  $y = u - v$ .
- Then  $(u + v)^3 + (u - v)^3 + 1 = 0$ .
- This simplifies to  $2u^3 + 6uv^2 + 1 = 0$ .
- Which leads to  $6(v/u)^2 = -(1/u)^3 - 2$ .
- So  $X = -6/u$ ,  $Y = 36v/u$ .
- Hence  $Y^2 = X^3 - 432$ .

## Weierstrass Equation

A two-variable equation  $F(x, y) = 0$ , forms a curve in the plane.

The generalized Weierstrass Equation of elliptic curves:

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

## Quadratic Equation

- $x^2 + ax + b = 0$
- $x = t - \frac{a}{2}$
- $t^2 - \frac{a^2}{4} - 4b = 0$

## Cubic Equation

- $x^3 + ax^2 + bx + c = 0$
- $x = t - \frac{a}{3}$
- $t^3 + pt + q = 0$
- $p = b - \frac{a^2}{3}$
- $q = c + \frac{2a^3}{27} - \frac{ab}{3}$

## Field Characteristics

- If characteristic field is not 2:

$$\left(v + \frac{a_1x}{2} + \frac{a_3}{2}\right)^2 = x^3 + \left(\frac{a_1^2}{4} + a_2\right)x^2 + a_4x + \left(\frac{a_1a_3}{4} + a_6\right)$$

$$\Rightarrow y_1^2 = x^3 + a'_2x^2 + a'_4x + a'_6$$

- If characteristics of field is neither 2 nor 3:

$$x_1 = x + \frac{a_2}{3}$$

$$\Rightarrow y_1^2 = x_1^3 + \Delta x + B$$

## Discriminant

- Discriminant of  $x^2 + bx + c$  is  $b^2 - 4c$
- $b^2 - 4c$  is non-zero  $\Leftrightarrow$  no double roots
- Discriminant of  $x^3 + ax + b$  is  $-4a^3 - 27b^2$
- $-4a^3 - 27b^2$  is non-zero  $\Leftrightarrow$  no double roots

## j-invariant

- Define  $j$  of this elliptic curve  $E$  as  $j(E)/1728 = 4a^3/(4a^3 + 27b^2)$
- If we change  $x = m^2x, y = m^3y$ , get  $\tilde{E}$ :
- then  $j(E) = j(\tilde{E})$
- $j$ -value fixes  $E$

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



## j-invariant

- If we change  $x = m^2x, y = m^3y$ , get  $\tilde{E}$ :
- then  $j(E) = j(\tilde{E})$
- Why not something like  $x = mx + ny^2 + s$ ?
- It has to keep the point at infinity and keep the form  $y^2 = x^3 + ax + b$

## Points on the Elliptic Curve (EC)

- Elliptic Curve over field  $L$
- $E(L) = \{\infty\} \cup \{(x, y) \in L \times L \mid y^2 + \dots = x^3 + \dots\}$
- It is useful to add the point at infinity.

## Group Law

- A group law may be defined where the sum of two points is the reflection across the x-axis of the third point on the same line
- Chords and tangents

## The Abelian Group

Given two points  $P, Q$  on  $E$ , there is a third point, denoted by  $P + Q$  on  $\tilde{E}$ , and the following relations hold for all  $P, Q, R$  in  $E$ .

- $P + Q = Q + P$  (commutativity)
- $(P + Q) + R = P + (Q + R)$  (associativity)
- $P + O = O + P = P$  (existence of an identity element)
- there exists  $(-P)$  such that  $(-P) + P = O$  (existence of inverses)

## Associativity

- $(P + Q) + R = P + (Q + R)$
- Associativity is non-trivial.
- It gives Pascal's theorem and Pappus's theorem.

## Elliptic Curve Picture

- Consider elliptic curve  $E : y^2 = x^3 - x + 1$
- If  $P_1$  and  $P_2$  are on  $E$ , we can define  $P_3 = P_1 + P_2$  as shown in the picture.

## Doubling of a point

- Let  $P = Q$
- $2y_1 \frac{dy}{dx} = 3x_1^2 + a$
- $m = \frac{dy}{dx} = \frac{3x_1^2 + a}{2y_1}$
- If  $y_1 \neq 0$  (since then  $P_1 + P_2 = \infty$ ):
  - $0 = x^3 - m^2x^2 + \dots$
  - $x_3 = m^2 - 2x_1, y_3 = m(x_1 - x_3) - y_1$
- What happens when  $P_2 = \infty = O$ ?

## Sum of two points

Define for two points  $P(x_1, y_1)$  and  $Q(x_2, y_2)$  in the Elliptic curve:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{for } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{for } x_1 = x_2 \end{cases}$$

Then  $P + Q$  is given by  $R(x_3, y_3)$ :

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_3 - x_1) + y_1 \end{aligned}$$

## What is -P?

- $y^2 = x^3 + ax + b$
- $P = (x_1, y_1)$
- What is -P? Is  $-P = (x_1, -y_1)$ ?
- Yes. But this works only for  $y^2 = x^3 + ax + b$ .
- For  $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$
- $-P = (x_1, -a_1x_1 - a_3 - y_1)$

## Motivation

- over  $\mathbb{F}_3$
- $Y^2Z + 2XYZ + YZ^2 = X^3 - XZ^2 + 7Z^3$  has a solution (1,2,1).
- Note that (0,1,0) is a solution.
- Important Point 1: We do not say (0,0,0) is a solution of the Weierstrass equation.

## Homogeneous vs Affine

- Important Point 2: We treat  $(1, 2, 1) \sim (2, 1, 2)$ , i.e., consider them to be identical and call it a point of the curve given by the Weierstrass equation.
- $\frac{5^2}{13^2} + \frac{12^2}{13^2} = \frac{13^2}{13^2}$
- $\frac{10^2}{26^2} + \frac{24^2}{26^2} = \frac{26^2}{26^2}$
- $X^2 + Y^2 = Z^2$  implies  $\left(\frac{X}{Z}\right)^2 + \left(\frac{Y}{Z}\right)^2 = 1$

## Projective Co-ordinates

- Two-dimensional projective space  $P_K^2$  over  $K$  is given by the equivalence classes of triples  $(x, y, z)$  with  $x, y, z$  in  $K$  and at least one of  $x, y, z$  non-zero.
- Two triples  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$  are said to be equivalent if there exists a non-zero element  $\lambda$  in  $K$ , such that:

$$(x_1, y_1, z_1) = (\lambda x_2, \lambda y_2, \lambda z_2)$$

- The equivalence class depends only on the ratios and hence is denoted by  $(x : y : z)$ .

## Singularity

- For an elliptic curve  $y^2 = f(x)$ , define  $F(x, y) = y^2 - f(x)$ . A singularity of the EC is a point  $(x_0, y_0)$  such that:
  - $\frac{\partial F}{\partial x}(x_0, y_0) = \frac{\partial F}{\partial y}(x_0, y_0) = 0$
  - or,  $2y_0 = -f'(x_0) = 0$
  - or,  $f(x_0) = f'(x_0)$
  - Therefore,  $f$  has a double root.

## Singularity

- $y^2 = x^2(x - 1)$  double roots  $x = 0$
- Let  $x - 1 = s^2$
- $y^2 = (s^2 + 1)^2(s^2)$
- Hence  $x = s^2 + 1, y = s(s^2 + 1)$

## If singular, then

- $K$  = a field
- $K(x, y) = K(t)$
- For  $y^2 = x^2(x - 1)$ ,  $x = s^2 + 1$ ,  $y = s(s^2 + 1)$
- For  $y^2 = x^3$ ,  $y = t^3$ ,  $x = t^2$
- For an elliptic curve,  $K(x, y)$  is never  $K(t)$ .

## Projective Form

- $E : Y^2Z + a_1XYZ + a_3Y^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$
- has a point  $(0, 1, 0)$ , point at infinity, denoted by  $O$ .

## Elliptic Curves in Characteristic 2

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

- If  $a_1$  is not 0, this reduces to the form:

$$y^2 + xy = x^3 + Ax^2 + B$$

- If  $a_1$  is 0, the reduced form is:

$$y^2 + a_3y = x^3 + Bx + C$$

- Note that the form cannot be:

$$y^2 = x^3 + Ax + B$$

## EC over Finite Fields

- An elliptic curve may be defined over any finite field  $GF(q)$ .
- For  $GF(2^m)$ , the curve has a different form:

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

- where  $b$  is not 0.
- Addition formulae are similar to those over the reals.

## Terminology

- Order of point  $P$  is the smallest integer  $r$  such that  $[r]P = O$ .
- Order of the curve is the number of points of  $E(\mathbb{F})$ , denoted by  $\#E(\mathbb{F})$ .

## Group Properties

- Let  $\#E(\mathbb{F}_q)$  denote the number of points on an elliptic curve  $E(\mathbb{F}_q)$ , including  $\mathcal{O}$ .
- Hasse bound:  $\#E(\mathbb{F}_q) = q + 1 - t$ , where  $|t| < 2\sqrt{q}$ .
- The group of points is either cyclic or a product of two cyclic groups.

## So it's an Abelian Group...

- Group homomorphism? Isogeny, isogenous.
- Endomorphism, isomorphic.
- Examples of endomorphisms are:
  - $[2] : E \rightarrow E, P \mapsto [2]P$
  - $[n] : E \rightarrow E, P \mapsto [n]P$

## Non-trivial Isogeny

- $E : y^2 = x^3 - x$
- $[i = \sqrt{-1}] : (x, y) \mapsto (-x, iy)$
- $[i = \sqrt{-1}]^2 = [i][i] = (-1) : (x, y) \mapsto (x, -y), P \mapsto -P$
- here  $i^2 = -1$
- $6^2 = -1 \pmod{37}$
- Called complex multiplication.

## Frobenius Map

- $\text{GF}(q)$ ,  $q = p^k$
- $F : \text{GF}(q) \rightarrow \text{GF}(q)$
- $F(x) = x^p$  for any  $x$
- $F$  is an isomorphism of  $\text{GF}(q)$ . So  $F$  defines an isogeny for any elliptic curve over  $\text{GF}(q)$ .

## $E[n]$

- For any group  $G$ , any natural number  $n$ ,  $G[n] = \{g \mid g^n = 1\}$ .
- $E[n] = \{P \mid [n]P = \mathcal{O}\}$ .

# Appendix A

## Additional Data A

### A.1 Existence of an Additional Root in Cubic Functions via the Intermediate Value Theorem

**Theorem:** Let  $f(x) = ax^3 + bx^2 + cx + d$  be a cubic function, where  $a, b, c, d \in \mathbb{R}$  and  $a \neq 0$ . If  $x_1$  and  $x_2$  are two distinct roots of  $f(x)$ , there exists at least one other root  $x_3$  of  $f(x)$ .

**Proof:**

1. *Cubic Function:* A cubic function is defined as  $f(x) = ax^3 + bx^2 + cx + d$ , which is a polynomial of degree 3, and thus continuous over  $\mathbb{R}$ .
2. *Known Roots:* Assume  $x_1$  and  $x_2$  are two distinct roots of  $f(x)$ , i.e.,  $f(x_1) = f(x_2) = 0$ .
3. *Intermediate Value Theorem (IVT):* The IVT states that for any continuous function  $g$  on an interval  $[a, b]$ , if  $g(a)$  and  $g(b)$  have opposite signs, there is at least one  $c$  in  $(a, b)$  such that  $g(c) = 0$ .
4. *Application to Cubic Function:* By the nature of cubic functions, they must change direction at least once between two roots. This implies the function will either attain a local maximum or minimum between  $x_1$  and  $x_2$ .
5. *Existence of Third Root:* If the local extremum is above or below the x-axis, the function must cross the x-axis to change direction, implying the existence of another root  $x_3$  in the interval  $(x_1, x_2)$ .
6. *Conclusion:* Therefore, by drawing a straight line through  $(x_1, 0)$  and  $(x_2, 0)$ , this line will intersect the graph of  $f(x)$  at least at one other point, indicating the existence of another root  $x_3$ .

