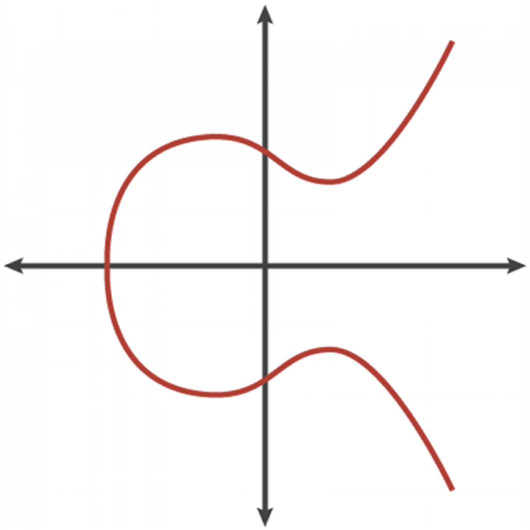
Elliptic Curve Cryptography

- majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials.
- imposes a significant load in storing and processing keys and messages
- an alternative is to use elliptic curves.
- offers same security with smaller bit sizes.



Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

Elliptic Curve Cryptography

- ECC use the two fields of EC ,prime i.e Z_P and 2 power of positive integer i.e GF(2^m) and also use two EC operation point addition and point doubling operation.

Some definitions

- Affine point: point present in EC
- O point : point of infinity(not present in EC)
- Generator point , G: point of the curve generate a secrete subgroup by repeating addition of G.
- Ord(G),n: number of point in the subgroup
- Cofactor, h:# of point in EC divided by number of point in subgroup .
- Its ideally 1 (i.e fields is prime).

Real Elliptic Curves

- An elliptic curve is defined by an equation in two variables x & y, with coefficients.
- consider a cubic elliptic curve of form
  - $y^2 = x^3 + ax + b$
    - where x,y,a,b are all real numbers
    - also define zero point O

Finite Elliptic Curves

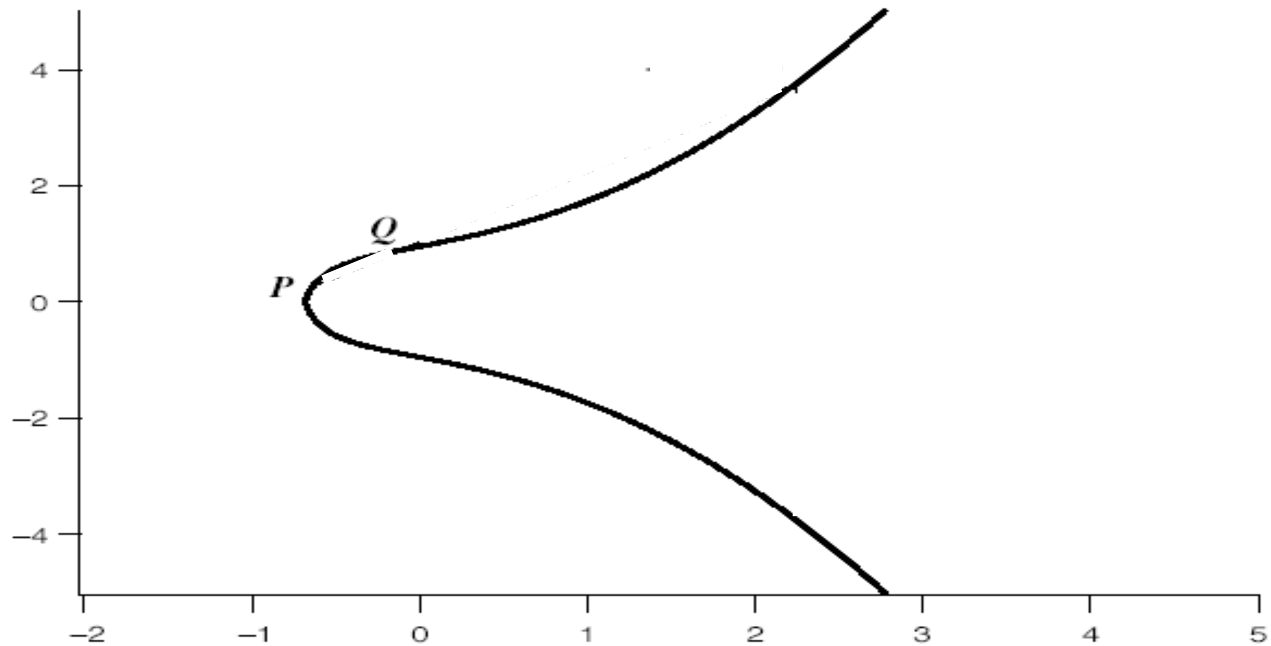
- Elliptic curve cryptography uses curves whose variables & coefficients are finite integers.
- have two families commonly used:
  - prime curves $E_p(a,b)$ defined over Z_p (Finite fields)
    - $y^2 \bmod p = (x^3+ax+b) \bmod p$
    - use integers modulo a prime for both variables and coeff
  - Example: $P=(3,10)$, $Q=(9,7)$, in $E_{23}(1,1)$
    - $P+Q = (17,20)$
    - $2P = (7,12)$

Real Elliptic Addition

Example: Points P=(3,10), and Q (9,7) in E_{23} (1,1)

Then, $E_p\,(a,b) \mid y^2 = x^3 + ax + b\ (mod\ p)$ will be

$$E_{23}(1,1) \mid y^2 = x^3 + x + 1\ (mod\ 23)$$

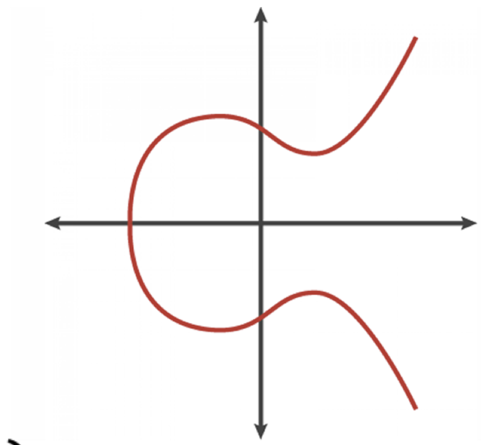


(b) $y^2 = x^3 + x + 1$

All points on $E_p(a,b)$

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

$$y = \pm\sqrt{x^3 + ax + b}$$



$E_{23}(1,1)$ → $y^2 = x^3 + x + 1\ (mod 23)$

$x, y = 0,1,2,3, \ldots \ldots \ldots \ldots \ldots \ldots (p-1)$

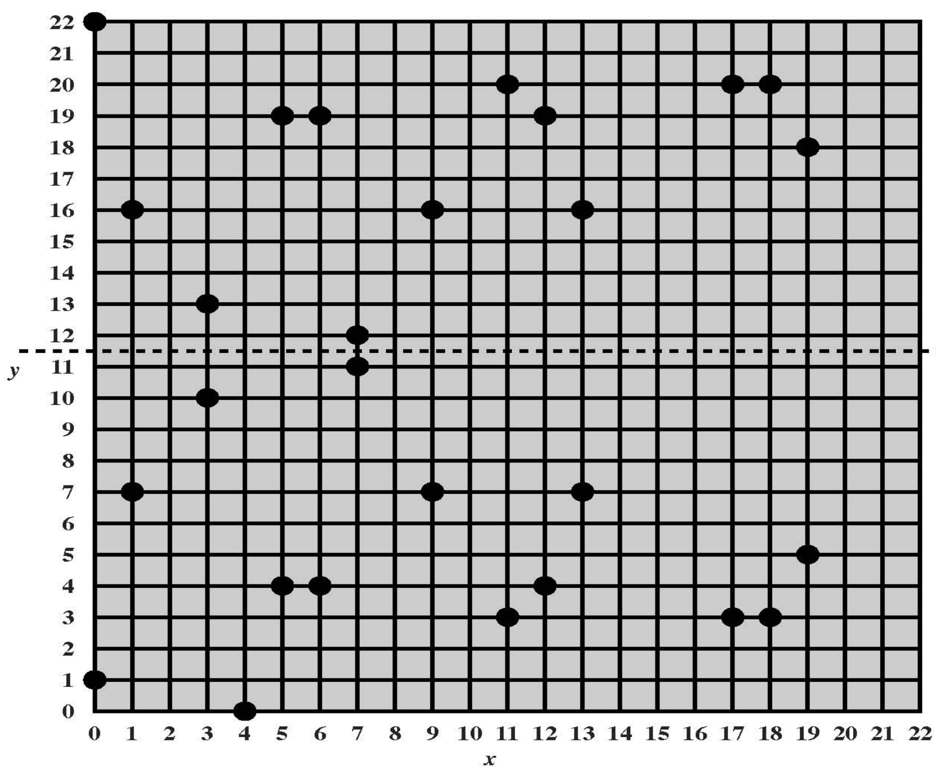


Figure 10.10 The Elliptic Curve $E_{23}(1,1)$

All points on $E_{23}(1,1)$

x	RHS
0	1
1	3
~~2~~	~~11~~
3	8
4	0
5	16
6	16
7	6
~~8~~	~~15~~
9	3
~~10~~	~~22~~
11	9
12	16
13	3
~~14~~	~~22~~
~~15~~	~~10~~
~~16~~	~~19~~
17	9
18	9
19	2
~~20~~	~~17~~
~~21~~	~~14~~
~~22~~	~~22~~

y	LHS
0	0
1	1
2	4
3	9
4	16
5	2
6	13
7	3
8	18
9	12
10	8
11	6
12	6
13	8
14	12
15	18
16	3
17	13
18	2
19	16
20	9
21	4
22	1

$$E_{23}(1,1) \longrightarrow \underbrace{y^2}_{\text{LHS}} = \underbrace{x^3 + x + 1}_{\text{RHS}} \ (mod 23)$$

$$x, y = 0,1,2,3, \dots \dots \dots \dots \dots \dots (p-1)$$

(0,1) (0,22)

(1,7) (1,16)

(3,10) (3,13)

(4,0)

(5,19) (5,4)

(6,19) (6,4)

(7,11) (7,12)

(9,7) (9,16)

(11,20) (11,3)

(12, 19) (12,4)

(13,7)(13,16)

(17,3) (17,20)

(18,20) (18,3)

(19,5) (19,18)

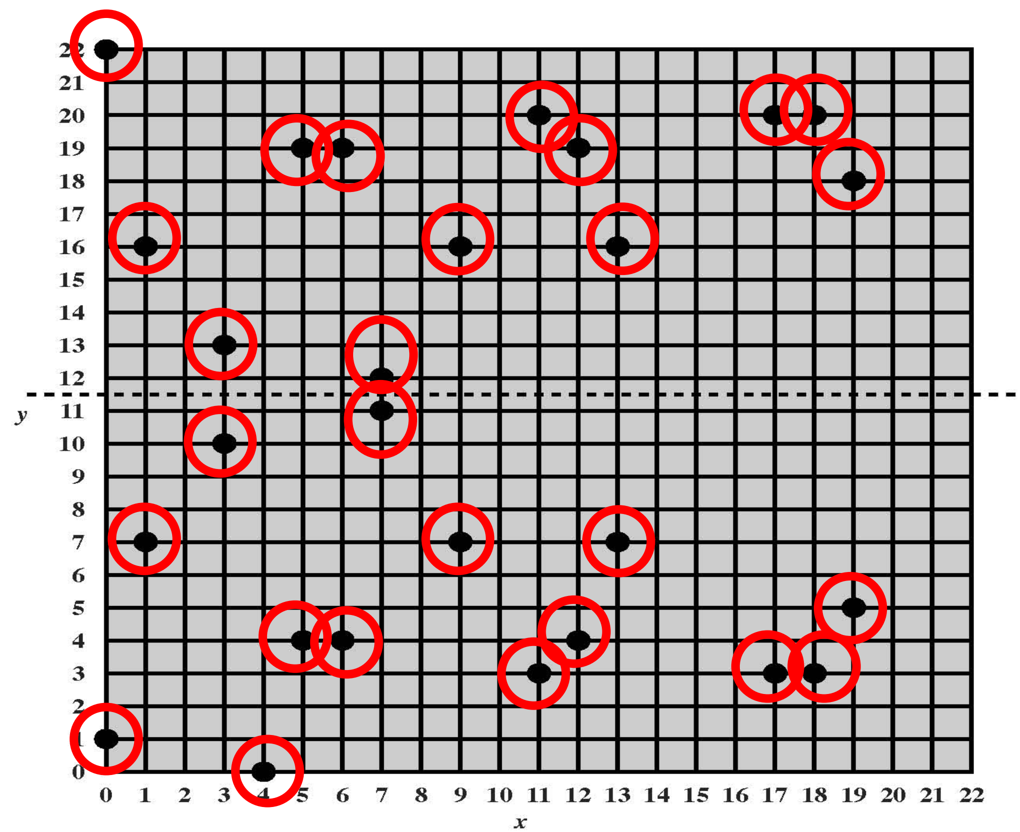


Figure 10.10 The Elliptic Curve $E_{23}(1,1)$

All points on $E_{23}(1,1)$

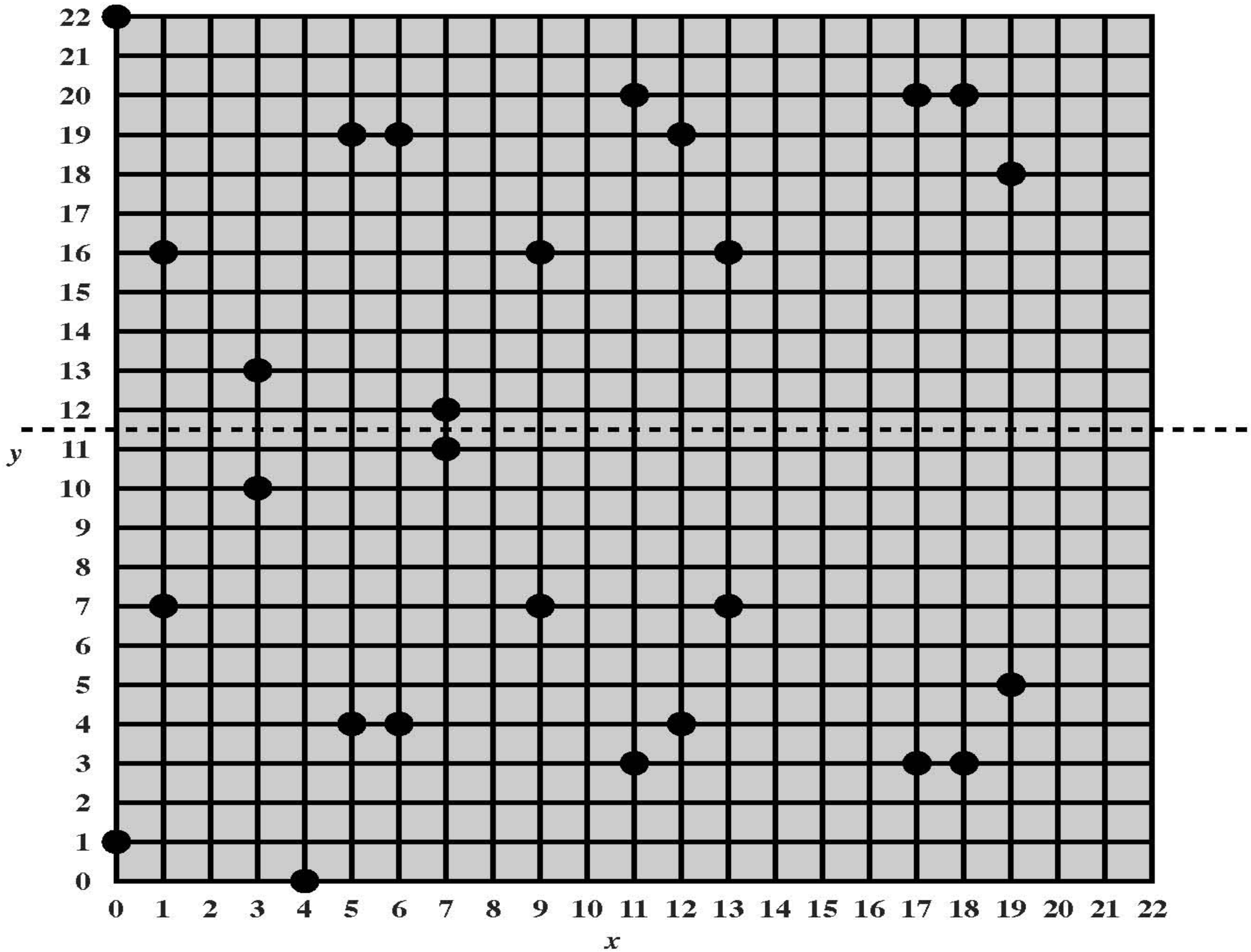
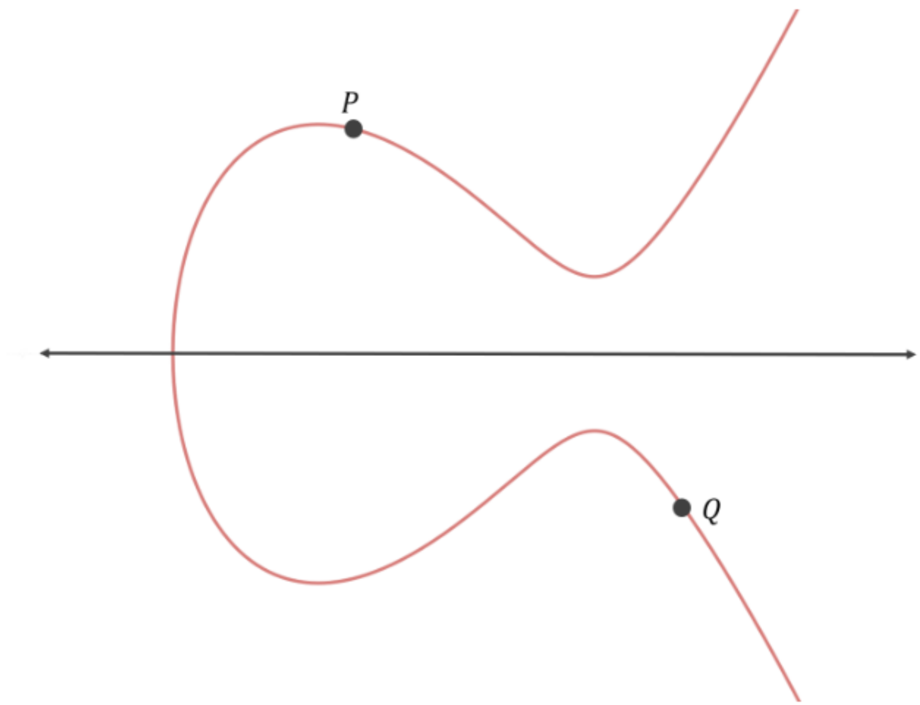
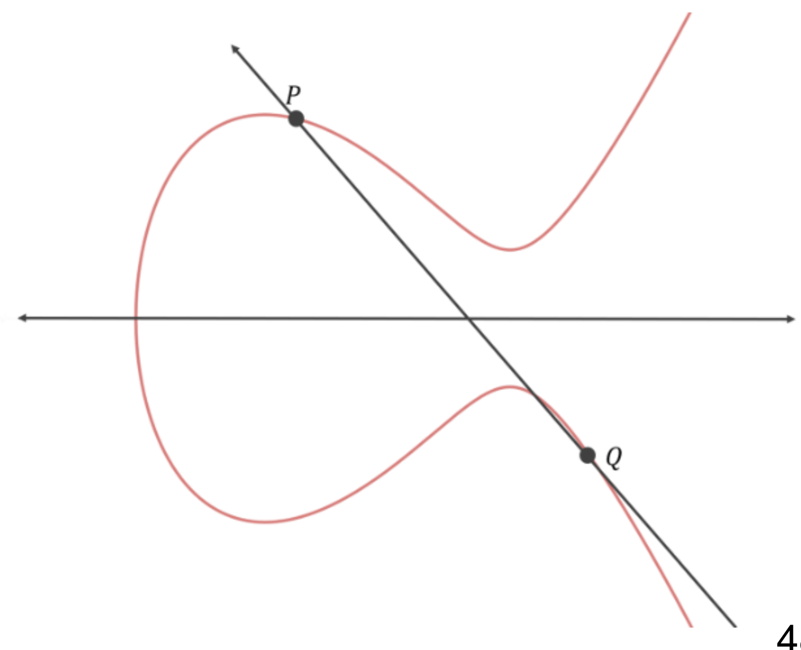


Figure 10.10 The Elliptic Curve $E_{23}(1,1)$

Point addition

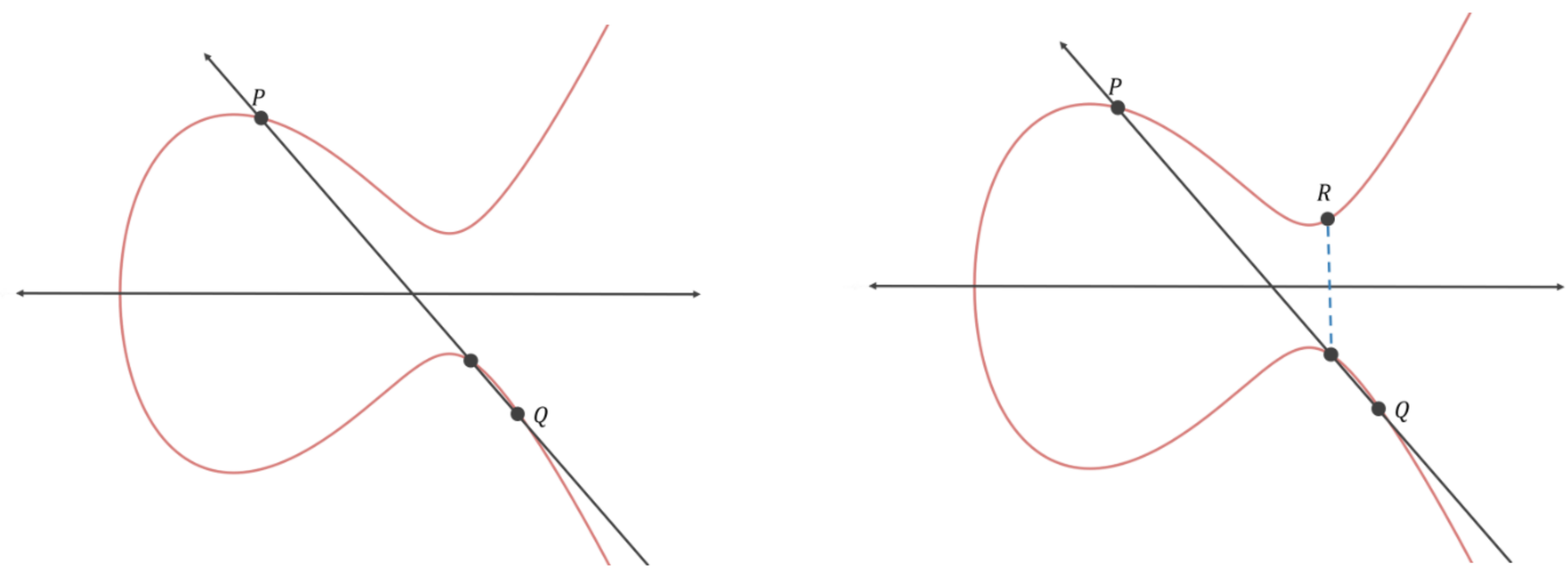
- You can add two points on an elliptic curve together to get a third point on the curve.
- To add two points on an elliptic curve together, you first find the line that goes through those two points.



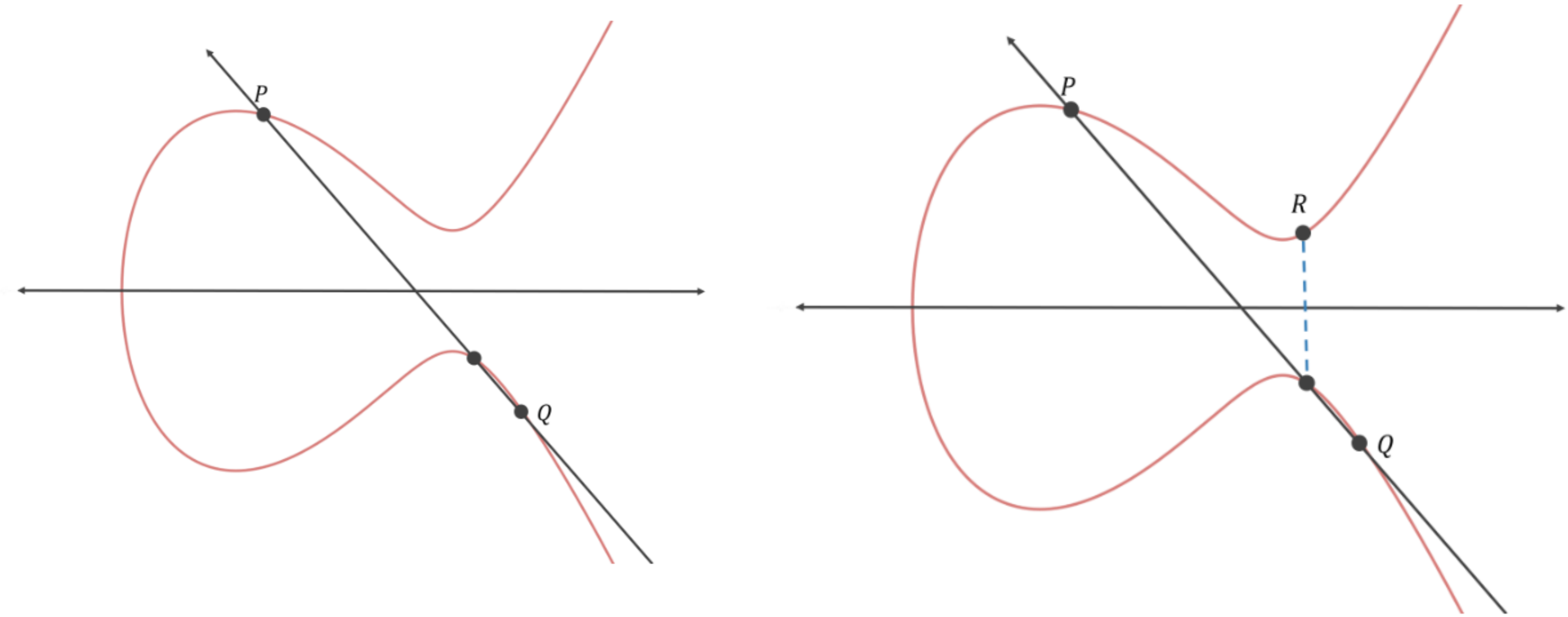


Point addition (Cont…)

- Then you determine where that line intersects the curve at a third point.
- Then you reflect that third point across the x-axis (i.e. multiply the y-coordinate by -1) and whatever point you get from that is the result of adding the first two points together.



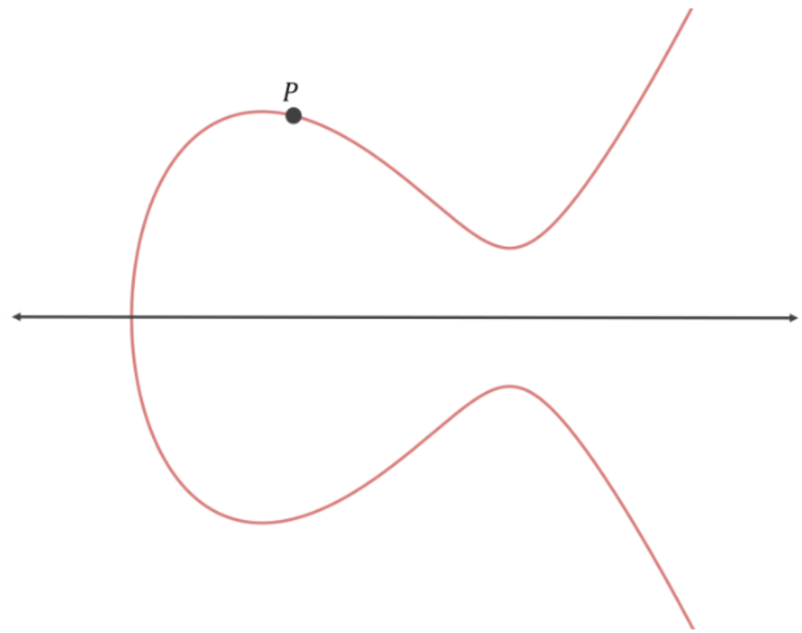
Point addition (Cont…)



- Then you reflect that point across the x-axis.
- Therefore, P+Q=R.

Point addition (Cont…)

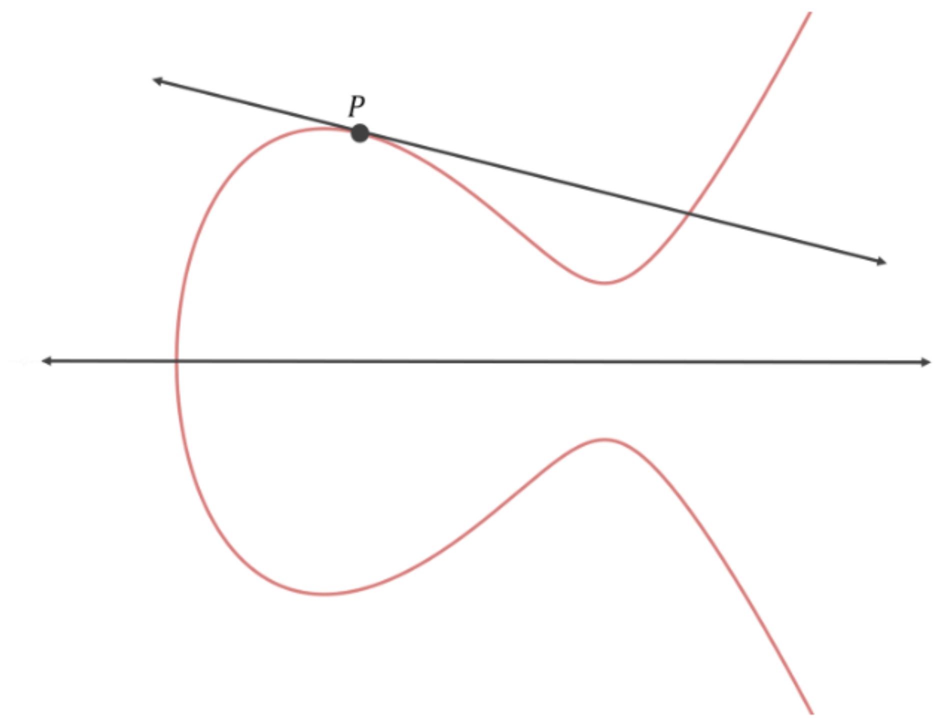
- To do elliptic curve cryptography properly, rather than adding two arbitrary points together, we specify a base point on the curve and only add that point to itself.
- For example, let’s say we have the following curve with base point P:



- Initially, we have P, or 1•P.

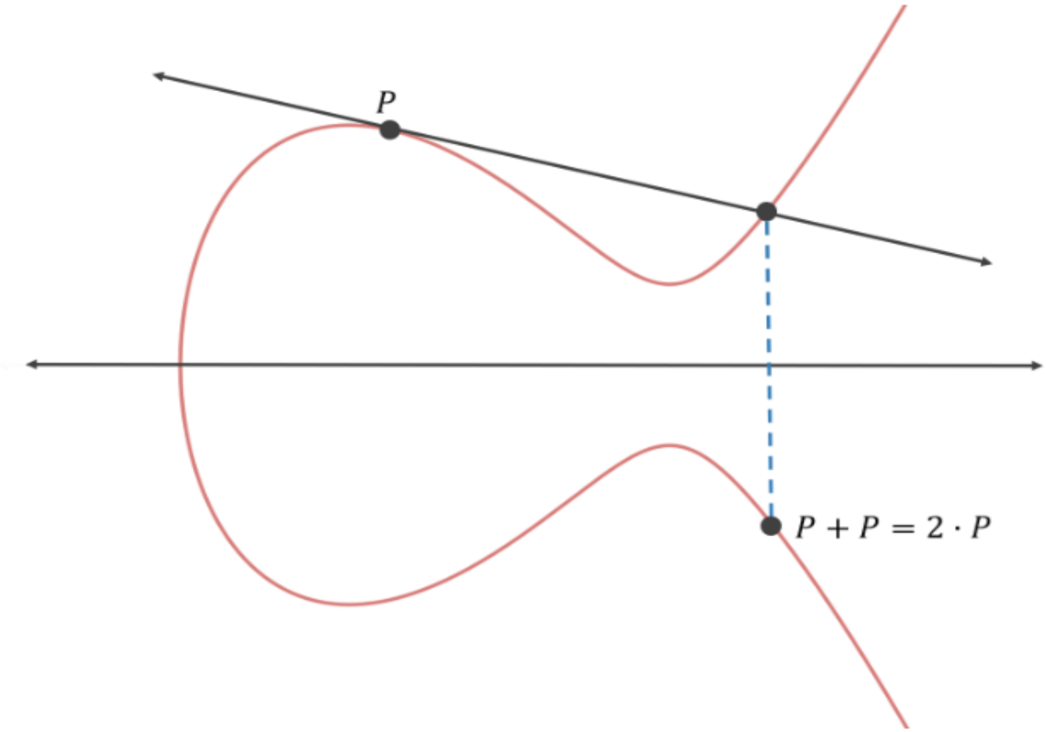
Point addition (Cont…)

- Now let's add P to itself.
- First, we have to find the equation of the line that goes through P and P.
- There are infinite such lines! In this special case, we opt for the tangent line.



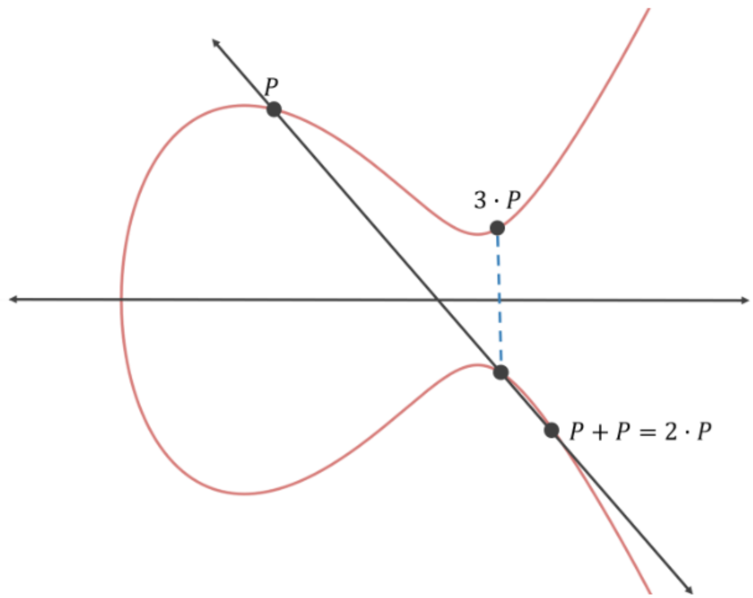
Point addition (Cont…)

- Now we find the “third” point that this line intersects and reflect it across the x-axis.
- Thus P added to itself, or P+P, equals 2•P.



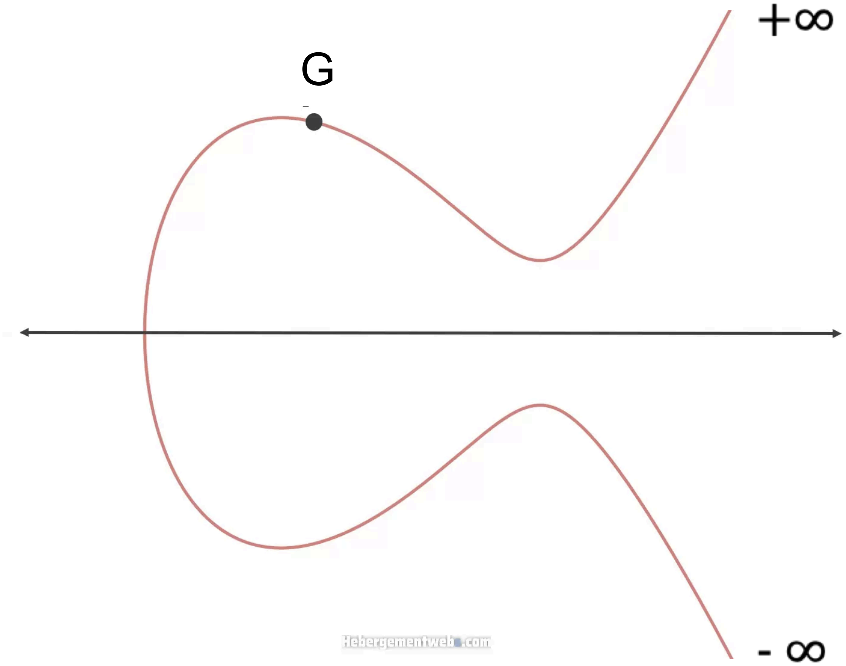
Point addition (Cont…)

- If we add P to itself again, we'll be computing P added to itself added to itself, or P+P+P. The result will be 3•P.
- To compute 3•P, we can just add P and 2•P together.



- We can continue to add P to itself to compute 4•P and 5•P and so on.

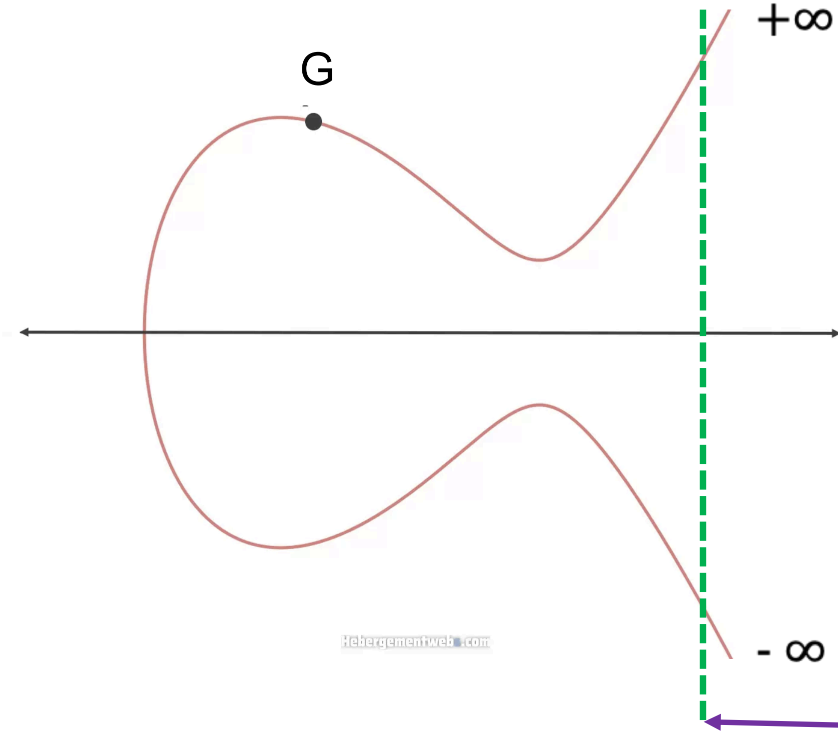
Subgroup Generation



Generator point, G:

For example, $E_{17}(2,2)$ => $y^2 = x^3$+2x+2 (mod 17) is G(5,1).

Subgroup Generation



n=19

$E_{17}(2,2)$ => $y^2 = x^3$+2x+2 (mod 17) is G(5,1).

- The **subgroup** of G calculated by repeated addition is Given below

G = (5,1)	6G = (16,13)	11G = (13,10)	16G = (10,11)
2G = (6,3)	7G = (0,6)	12G = (0,11)	17G = (6,14)
3G = (10,6)	8G = (13,17)	13G = (16,4)	18G = (5,16)
4G = (3,1)	9G = (7,6)	14G = (9,1)	19G = O
5G = (9,16)	10G = (7,11)	15G = (3,16)	

How to calculate 2G, 3G?

Generator point, G:

For example, $E_{17}(2,2)$ => y^2 = x^3+2x+2 (mod 17) is G(5,1).

G = (5,1) = (x_g, y_g); $(a, b) \equiv (2, 2)$

- **2G = G+G (called point doubling operation)=(x_{2g}, y_{2g})**
  - $$x_{2g} = \left(\frac{3x_g^2+a}{2y_g}\right)^2 - 2x_g$$
  - $$y_{2g} = \left(\frac{3x_g^2+a}{2y_g}\right)\left(x_g - x_{2g}\right) - y_g$$
- **3G = G+2G (called point addition operation)=(x_{3g}, y_{3g})**
  - $$x_{3g} = \left(\frac{y_{2g}-y_g}{x_{2g}-x_g}\right)^2 - x_g - x_{2g}$$
  - $$y_{3g} = \left(\frac{y_{2g}-y_g}{x_{2g}-x_g}\right)\left(x_g - x_{2g}\right) - y_g$$

How to calculate 2G, 3G?

For calculating 2G:

Now, $x_{2g} = s^2 - 2x_g$

$= 13^2 - 2*5 \pmod{17}$

$= 16 – 10 \pmod{17}$

$= 6$

and, $y_{2g} = s(x_g - x_{2g}) - y_g$

$= 13(5 - 6) – 1$

$= -13 – 1 = -14 \pmod{17}$

$= 3$

so, 2G = (6,3)

Let

$$s = \frac{3x_g{}^2 + a}{2y_g}$$

$$= \frac{3*5^2+2}{2(1)}$$

$= 77* \pmod{17}$

$= 9*9 \pmod{17}$

$= 13$

$$\boldsymbol{x_{2g}} = \left(\frac{3x_g{}^2 + a}{2y_g}\right)^{\mathbf{2}} - \boldsymbol{2x_g}$$

$$= \boldsymbol{s^2 - 2x_g}$$

$$\boldsymbol{y_{2g}} = \left(\frac{3x_g{}^2 + a}{2y_g}\right)\boldsymbol{(x_g - x_{2g}) - y_g}$$

$$= \boldsymbol{s(x_g - x_{2g}) - y_g}$$

How to calculate 2G, 3G?

For calculating 3G:

3G = G + 2G = (5,1) + (6,3)

now, $x_{3g} = s^2 - x_g - x_{2g}$

$= 2^2 - 5 - 6$

$= -7 \pmod{17}$

$= 10$

and ,

$y_{3g} = s\,(x_g - x_{3g}) - y_g$

$= 2\,(5 - 10) - 1$

$= -11 \pmod{17}$

$= 6$

so, 3G = (10,6)

Let

$$s = \frac{y_{2g} - y_g}{x_{2g} - x_g}$$

$$= \frac{3-1}{6-5}$$

$$= 2$$

$$x_{3g} = \left(\frac{y_{2g} - y_g}{x_{2g} - x_g}\right)^2 - x_g - x_{2g}$$

$$= s^2 - x_g - x_{2g}$$

$$y_{3g} = \left(\frac{y_{2g} - y_g}{x_{2g} - x_g}\right)(x_g - x_{2g}) - y_g$$

$$= s(x_g - x_{2g}) - y_g$$

Quick Task 1

Given, G = (5,1), 2G = (6,3), 3G = (10,6).

Find out the value of 4G using E_{17}**(2,2)**

We can do that by any of the following operations:

Point Doubling Operation

4G = 2G+2G.

$s = \frac{3x_{2g}^2 + a}{2y_{2g}}$

$x_{4g} = s^2 - 2x_{2g}$

$y_{4g} = s(x_{2g} - x_{4g}) - y_{2g}$

Point Addition Operation

4G = 3G+G.

$s = (y_{3g} - y_g)/(x_{3g} - x_g)$

$x_{4g} = s^2 - x_g - x_{3g}$

$y_{4g} = s(x_g - x_{4g}) - y_g$

4G = (3,1).

Key Exchange Using ECC

Bob

1. Private key, β

   $1 \leq \beta \leq n-1$

2. Compute PU, $P_B = \beta G$

3. Receives,

   $P_A = \alpha G = (x_{P_A}, y_{P_A})$

4. Computes,

   Key = $\beta(P_A)$

Global Public Elements

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

G, n

Point on ECC whose order is large value

a, b

P_A, P_B

Key = ?

Alice

1. Private key, α

   $1 \leq \alpha \leq n-1$

2. Compute PU, $P_A = \alpha G$

3. Receives,

   $P_B = \beta G = (x_{P_B}, y_{P_B})$

4. Computes,

   Key = $\alpha(P_B)$

Secret Key

Key Exchange Using ECC

Example

Bob	Eve (3rd party)	Alice
1. $Private\ key,$ $\beta = 9$	$\boldsymbol{E_{17}(2,2)}$ $\Rightarrow \boldsymbol{y^2 = x^3 + 2x + 2(mod\ 17)}$ $\mathbf{G = (5,1), n = 19}$	1. $Private\ key,$ $\alpha = 3$
2. ***Compute ,*** $P_B = \beta \mathrm{G} = 9\mathrm{G} = (7{,}6)$		2. ***Compute ,*** $P_A = \alpha\mathrm{G} = 3\mathrm{G} = (10{,}6)$
3. ***Recevies***, $P_A = (10{,}6)$	$\boldsymbol{P_B = (7,6),}$ $\boldsymbol{P_A = (10,6)}$	3. ***Recevies,*** $P_B = (7{,}6)$
4. ***Computes,*** K= $\beta P_A = \beta\alpha$G=9(3G) $= 9(10{,}6) = (13{,}17)$	$\alpha = P_A$/G=Infeasible $\beta = P_B$/G=Infeasible So, K=???	4. ***Computes,*** K=$\alpha P_B = \alpha\beta$G=3(9G) $= 3(7{,}6) = (13{,}17)$

Why ECC is more secure?

- Consider the equation $Q = kP$ where $Q, P \in E_p(a,b)$ and $k < p$.
- It is relatively easy to calculate Q given k and P, but it is way too much hard to determine k given Q and P.
- This is called the discrete logarithm problem for elliptic curves.
- Example: Consider the group $E_{23}(9,17)$. This is the group defined by the equation $y^2 \bmod 23 = (x^3+9x+17) \bmod 23$.
- Find out the value of k given $Q = (4,5)$ and the base $P = (16,5)$
- The brute-force method is to compute multiples of P until Q is found. Thus,
- P = (16,5); 2P = (20,20); 3P = (14,14); 4P = (19,20); 5P = (13,10); 6P = (7,3); 7P = (8,7); 8P = (12,17); 9P = (4,5) = Q.
- In a real application, k would be so large as to make the brute-force attack infeasible.

How to encrypt or decrypt using ECC?

- The first task in this system is to encode the plaintext message m to be sent as an x-y point P_m.
- It is the point P_m that will be encrypted as a cipher text and subsequently decrypted.
- Note that, we cannot simply encode the message as the x or y coordinate of a point, because not all such coordinates are in $E_k(a,b)$.

Encryption & Decryption using ECC

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

$$C_m = \{\alpha G, P_m + \alpha P_B\}$$

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

$$P_m + \alpha P_B - \beta(\alpha G) = P_m + \alpha(\beta G) - \beta(\alpha G) = P_m$$

Encryption & Decryption using ECC

- Example: $E_{17}(2,2)$ => $y^2 = x^3+2x+2 \pmod{17}$ and G(5,1)
- We consider (6,3) point on the EC as P_m.
- A selects $\alpha = 2$. B selects $\beta = 3$,
- Thus, $P_B = \beta G = 3G = (10,6)$.
- We have $\alpha G = 2G = (6,3)$, and

$$\{P_m + \alpha P_B\} = \{(6,3)+2(10,6)\}$$
$$= \{(6,3)+(16,13)\} = (13,7).$$

- Thus A sends the cipher text

$$C_m = \{(6,3),(13,7)\}$$

Quick task 2: Decrypt C_m and Check if it is P_m

ECC Encryption/Decryption

Quick task 2:

Decrypt C_m and Check if it is P_m