Elliptic Curve Cryptography

Elliptic Curve Cryptography

- Majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials
- It imposes a significant load in storing and processing keys and messages
- An alternative is to use elliptic curves
- It offers same security with smaller bit sizes
- Newer, but not as well analysed

Elliptic Curves over Real Numbers

- An elliptic curve is defined by an equation in two variables x & y, with coefficients
- For cryptography, the variables and coefficients are restricted to elements in a Finite field.

Consider an elliptic curve

- where x, y, a, b, the variables and coefficients are all real numbers
- In general, the cubic equations for elliptic curves takes the form

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

Elliptic Curves over Real Numbers

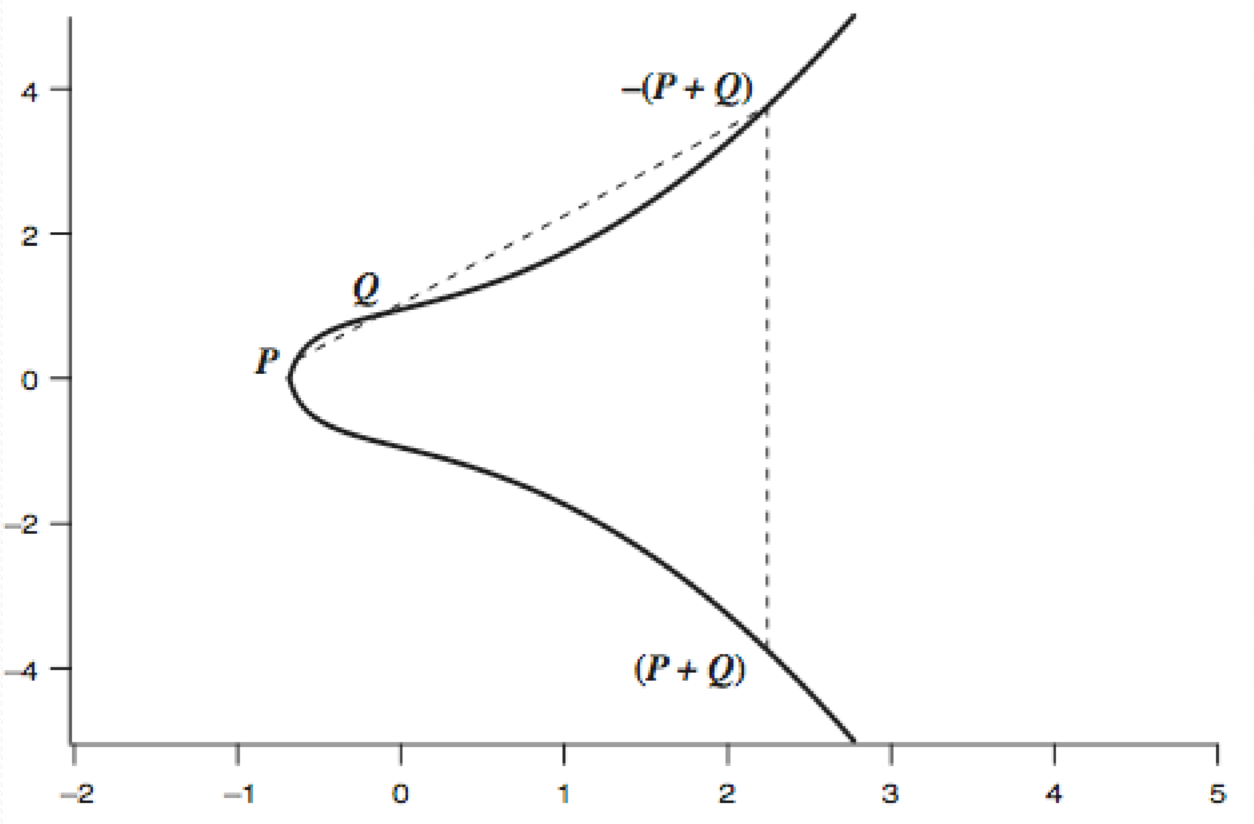
- Consider a cubic elliptic curve of form
  - $y^2 = x^3 + ax + b$
  - where x, y, a, b are all real numbers
  - also define zero point O or point at infinity
- consider set of points E(a,b) that satisfy

the equation $y = \sqrt{(x^3 + ax + b)}$

- Given a and b, the plot consists of positive and negative values of y for each value of x.
- Each curve is symmetric about y = 0

Real Elliptic Curve Example

geometrically sum of P+Q is reflection of the intersection R [= - (P+Q)]



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

Geometric Description of Addition

- A group can be defined based on the set E(a,b) provided that $x^3 + ax + b$ *has no repeated factors*

- Equivalent to the condition

$$4a^3 + 27b^2 \neq 0$$

- In geometric terms the rules for addition is

“ if three points on an elliptic curve lie on a straight line, their sum is o “

Rules for Addition

- 0 serves the additive identity

  $P + 0 = 0 + P = P$, assume $P \neq 0$ and $Q \neq 0$

- If $P = (x,y)$ then $-P = (x, -y)$. These two poits can be joined by a vertical line.

  $P + (-P) = P - P = 0$

- To add two points P and Q with different x co-ordinates, draw a straight line between them and find a third point of intersection R

  $P + Q = - R$

If the line is tangent to the curve at either P or Q, then R = P or R = Q.

Rules for Addition

- P and (-P), with same x-coordinate are joined by a vertical line, which can be viewed as

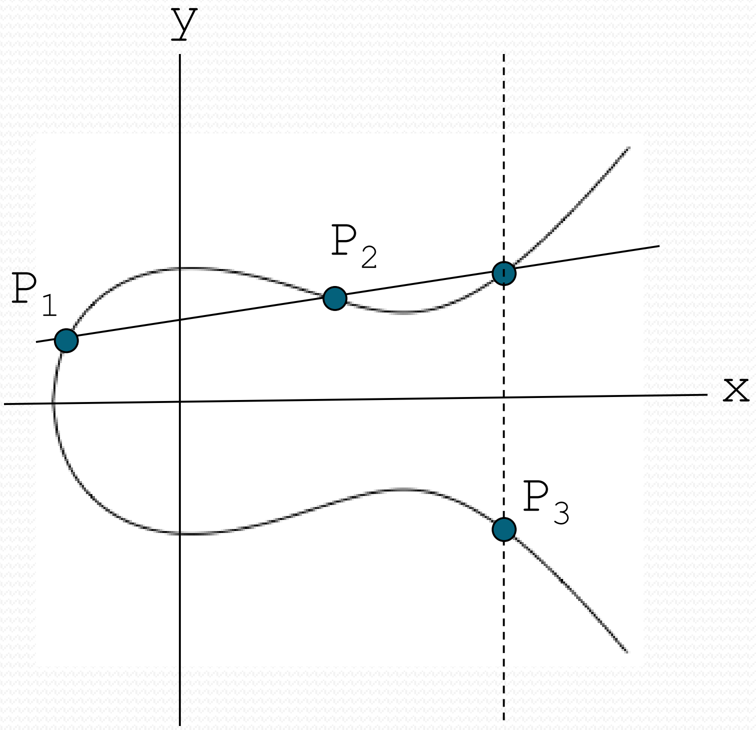
  intersecting the curve at the infinity point

  Therefore, $P + (-P) = 0$

- To double a point Q, draw the tangent line and find the other point of intersection S.

  Then $Q + Q = 2Q = -S$

Elliptic Curve Addition



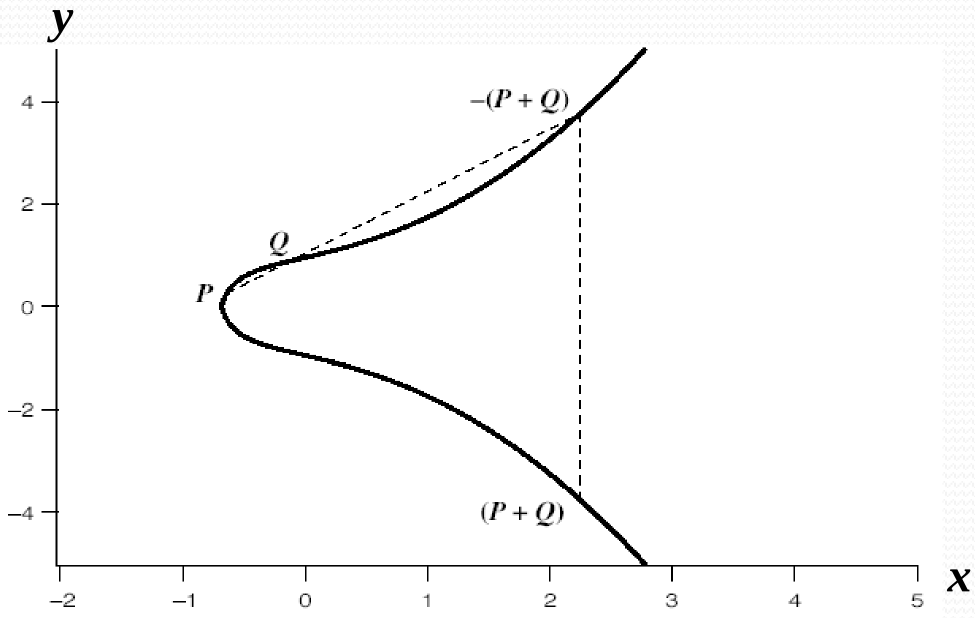
- Consider elliptic curve

$$E:\ y^2 = x^3 - x + 1$$

- If P_1 and P_2 are on E, we can define

$$P_3 = P_1 + P_2$$

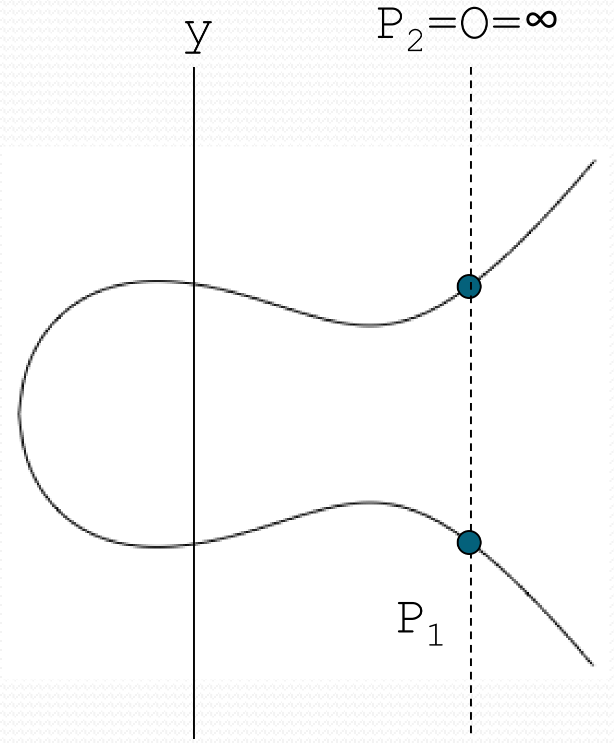
Addition in ECC



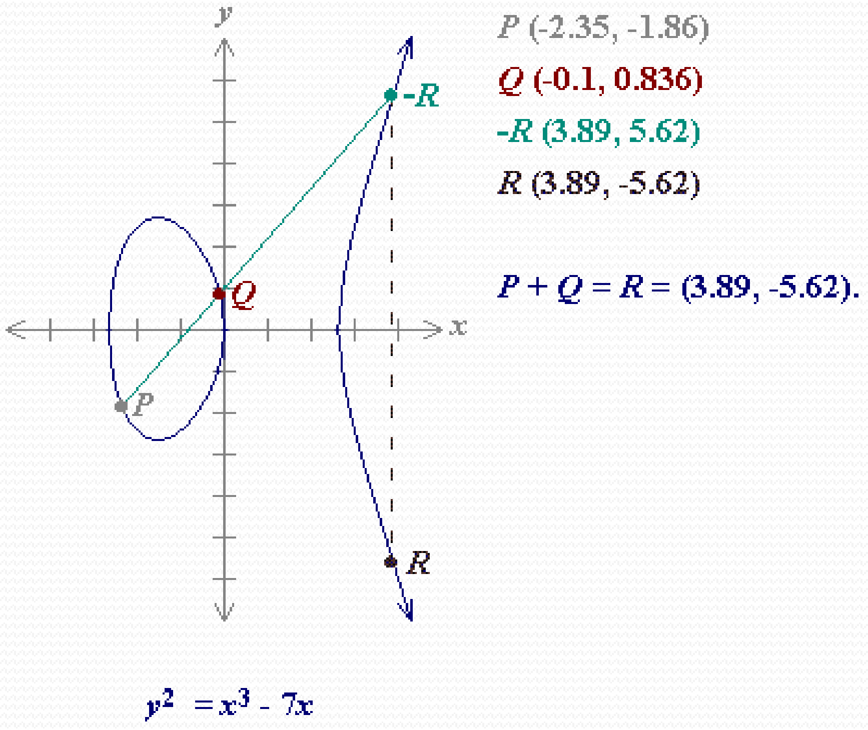
(b) $y^2 = x^3 + x + 1$

Let, P≠Q,

Addition

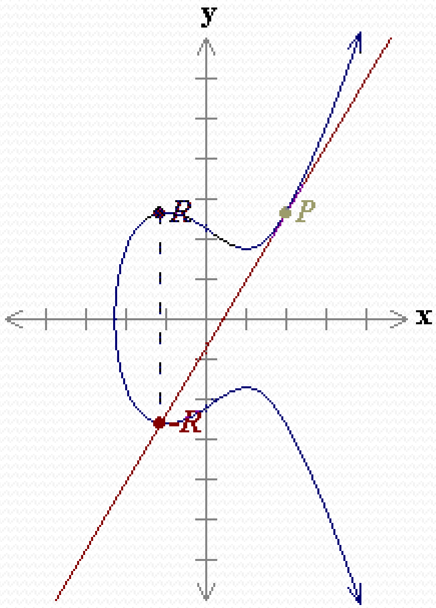


Addition



Point at infinity *O*

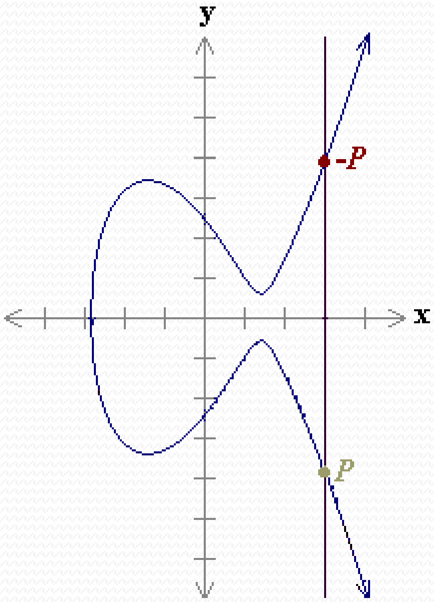
P+P = 2P



P (2, 2.65)
$-R$ (-1.11, -2.64)
R (-1.11, 2.64)

$2P = R = (-1.11, 2.64)$.

$y^2 = x^3 - 3x + 5$



$P + (-P) = O$

$y^2 = x^3 - 6x + 6$

As a result of the above case ***P=O+P***

O **is called the additive identity of the elliptic curve group.**

Hence all elliptic curves have an additive identity ***O***.

Finite Elliptic Curves

- Elliptic curve cryptography uses curves whose variables & coefficients are finite
- have two families commonly used:
  - prime curves $E_p(a,b)$ defined over Z_p
    - use integers modulo a prime
    - best in software
  - binary curves $E_{2m}(a,b)$ defined over $GF(2^n)$
    - use polynomials with binary coefficients
    - best in hardware

Elliptic Curve Cryptography

- ECC addition is analog of modulo multiply
- ECC repeated addition is analog of modulo exponentiation
- need “hard” problem equiv to discrete log
  - Q=kP, where Q,P belong to a prime curve
  - is “easy” to compute Q given k,P
  - but “hard” to find k given Q,P
  - known as the elliptic curve logarithm problem
- Certicom example: $E_{23}(9,17)$

ECC Diffie-Hellman

- can do key exchange analogous to D-H
- users select a suitable curve $E_q(a,b)$
- select base point $G=(x_1,y_1)$
  - with large order n s.t. $nG=O$
- A & B select private keys $n_A<n$, $n_B<n$
- compute public keys: $P_A=n_AG$, $P_B=n_BG$
- compute shared key: $K=n_AP_B$, $K=n_BP_A$
  - same since $K=n_An_BG$
- attacker would need to find k, hard

ECC Encryption/Decryption

- several alternatives, will consider simplest
- must first encode any message M as a point on the elliptic curve P_m
- select suitable curve & point G as in D-H
- each user chooses private key $n_A<n$
- and computes public key $P_A=n_AG$
- to encrypt P_m : $C_m=\{kG, P_m+kP_b\}$, k random
- decrypt C_m compute:

$$P_m+kP_b-n_B(kG) = P_m+k(n_BG)-n_B(kG) = P_m$$

ECC Security

- relies on elliptic curve logarithm problem
- fastest method is “Pollard rho method”
- compared to factoring, can use much smaller key sizes than with RSA, etc.
- for equivalent key lengths computations are roughly equivalent
- hence for similar security ECC offers significant computational advantages

Comparable Key Sizes for Equivalent Security

Symmetric scheme (key size in bits)	ECC-based scheme (size of *n* in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360