Basics of Elliptic Curves Cryptosystems

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@unipi.it
Version: 2021-04-18

ECC in a nutshell

- Mid-1980s
- Same level of security of RSA and DL-system with considerably shorter operands
  - 160 – 256 bit vs 1024 – 3072 bit
- Based on GDLP
  - DHKE and DL-systems can be realized using ECCs
- Performance advantages over RSA and DL-systems
  - RSA with short public parameter is faster than ECC

Elliptic Curves Cryptosystem

HOW TO COMPUTE WITH ECC

How to Compute with ECC

UNIVERSITÀ DI PISA

- ECC is based on GDLP so we have to accomplish two tasks
  - Task 1: Define a elliptic-curve-based cyclic group
    - Task 1.1: Define a set of elements
    - Task 1.2: Define the group operation
  - Task 2: Show that DLP is hard in that group

ELEMENTI

DEFINIRE $\underbrace{(G, *)}_{\text{GRUPPO}}$ OPERAZIONI

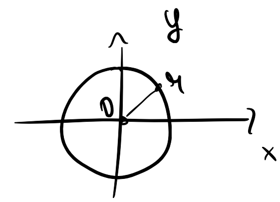
$(\mathbb{Z}_p^*, \times)$ MOLTIPLICAZIONE mod p

$\{1, 2 \ldots p-1\}$

Polynomials and curves

UNIVERSITÀ DI PISA

- We can form curves from polynomial equations
  - A curve is the set of points (x, y) which are the solutions of the equations
- Examples (in $\mathbb{R}$)
  - $x^2 + y^2 = r^2$ is a circle
  - $a \cdot x^2 + b \cdot y^2 = c$ is an ellipse



ECC – definition

UNIVERSITÀ DI PISA

- We consider GF(p) = {0, 1, …. p – 1 } GALOIS FIELD
  - Intuitively, GF is a finite set where you can add, subtract, multiply and invert
- Definition
  - The elliptic curve over $\mathbb{Z}_p$, p > 3, is the set of points $(x, y) \in \mathbb{Z}_p$ which fulfils

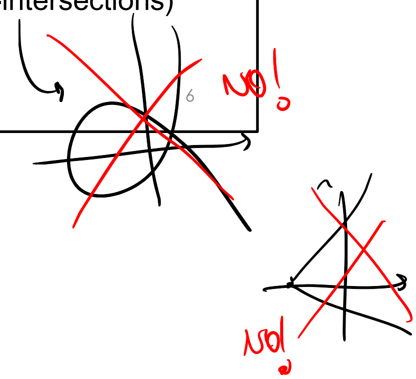
$$y^2 \equiv x^3 + a \cdot x + b \bmod p$$

SCEGLIENDO A E B CAMBIA LA CURVA

  - together with an imaginary point of infinity $\mathcal{O}$, where $a, b \in \mathbb{Z}_p$, and the condition

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

    - The curve is non-singular (no vertices, no self-intersections)

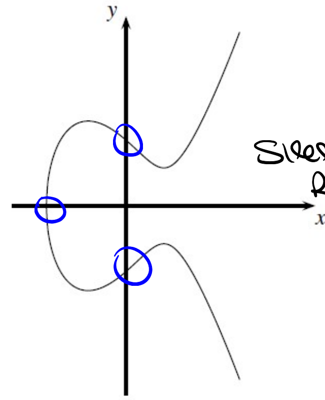


Group elements (task 1.1)

UNIVERSITÀ DI PISA

- Plotting in $\mathbb{R}$ for the sake of illustration
- Observations
  - 1, 3 intersections with x axis
  - Symmetric with respect to x axis
- Task 1.1 solved
  - Group elements are the points of the curve

CONSIDERO SOLO PUNTI!



SIMMETRICA RISPETTO x

$y^2 = x^3 - 3x + 3$ over $\mathbb{R}$

↳ PER POTERLA OSSERVARE

Group operations (task 1.2)

UNIVERSITÀ DI PISA

- We call "addition" the group operation and denote it by "+" an operation that takes two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ and produces a third point $R = (x_3, y_3)$ as a result

$$P + Q = R$$

ADDIZIONE TRA DUE PUNTI

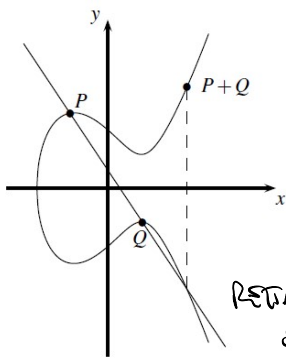
- Geometrical interpretation of + in $\mathbb{R}$
  - Point Addition $P + Q$, $Q \neq P$
  - Point Doubling $P + P$, $P = Q$

Group operations (task 1.2)

UNIVERSITÀ DI PISA

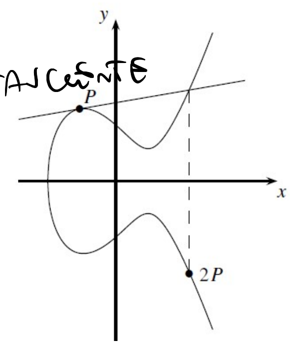
- Geometrical interpretation of "+" operation
  - The tangent-and-chord method

Point addition **Point doubling**



RETTA PER DUE PUNTI E SIMMETRICO

TANGENTE



INTERPRETAZIONE PER ℝ

Group operations (task 1.2)

- Geometrical interpretation of +
  - The tangent-and-chord method only uses the four standard operations
- FACT
  - If addition + is defined this way, the group points fulfil most of necessary conditions of a group: closure, associativity, existence of an identity element and existence of an inverse

→ NON DIMOSTRATO!

Group operations (task 1.2)

UNIVERSITÀ DI PISA

- Elliptic Curve Point Addition and Point Doubling
  - Analytic expressions
  - $x_3 \equiv s^2 - x_1 - x_2 \bmod p$
  - $y_3 \equiv s \cdot (x_1 - x_3) - y_1 \bmod p$
  - where
  - $s \equiv \frac{y_2 - y_1}{x_2 - x_1} \bmod p$ if P ≠ Q (point addition)
  - $s \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p$ if P = Q (point doubling)
  - with s the slope of chord/tangent

$P = (x_1, y_1)$

$Q = (x_2, y_2)$

$P + Q = (x_3, y_3)$

PENDENZA DELLA CURVA DELLA TANGENTE

Point at infinity (task 1.2)

UNIVERSITÀ DI PISA

- An identity (neutral) element $\mathcal{O}$ is still missing
  - $\forall P \in E$: $P + \mathcal{O} = P$
- There exists not such a point on the curve
- Thus, we define $\mathcal{O}$ as the point at infinity
  - Located at "plus" infinity towards the y-axis or at "minus" infinity towards the y-axis
- Now, we also define –P (inverse)
  $P + (-P) = \mathcal{O}$

POSTULATO L'ESISTENZA DI $\mathcal{O}$

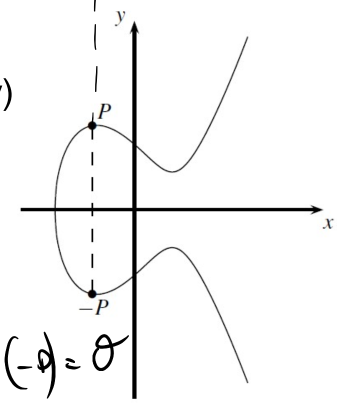
One thing that is still missing is an identity (or neutral) element $\mathcal{O}$.

Group operations (task 1.2)

$\mathcal{O}$

INFINITO

- Inverse of a point P on an elliptic curve
  - Apply the tangent-and-chord method
- In ECC over GF(p)
  - Given P = (x, y) then –P = (x, p – y)



$y_P + y_{-P} \bmod p = \emptyset$

$y + (p - y) = p \bmod p = \emptyset$

$P + (-P) = \mathcal{O}$

Elliptic Curves Cryptosystem

BUILDING DLP ON EC

A useful theorem

- THM
  - The points on an elliptic curve together with $\mathcal{O}$ have cyclic subgroups. Under certain conditions all points on an elliptic curve form a cyclic group
    - A primitive element must exist such that its powers generate the entire group

Example (1/2)

UNIVERSITÀ DI PISA

APPLICANDO LE EQUAZIONI E' POSSIBILE VEDERE CHE LA CURVA RAPPRESENTA UN SUBGROUP DI ORDINE 19

- E: $y^2 \equiv x^3 + 2 \cdot x + 2 \bmod 17$
  - #E (order of E) = 19
  - P = (5, 1) primitive element
  - “Powers” of P
    - 2P = (6, 3) – point doubling
    - 3P = (10, 6) – point addition 2P + P
    - 4P = (3, 1) - POINT DOUBLING 2×2P
    - 5P = (9, 16)
    - 6P = (16, 13)
    - 7P = (0, 6)
    - 8P = (13, 7)
    - 9P = (7, 6)
    - 10P = (7, 11)
    - 11P = (13, 10)
    - 12P = (0, 11)
    - 13P = (16, 4)
    - 14P = (9, 1)
    - 15P = (3, 16)
    - 16P = (10, 11)
    - 17P = (6, 14)
    - 18P = (5, 16)
    - 19P = $\mathcal{O}$ = #E·P

Example (2/2)

- The cyclic structure becomes visible
  - 20 P = 19P + P = O + P = P
  - 21P = 19P + 2P = 2P
  - …
- Furthermore
  - 19P = O, thus 18P + P = O, then 18P is the inverse of P and vice versa
    - Verification
      - P = (5, 1), 18P = (5, 16)
      - $x_p = x_{18P} = 5$
      - $y_p + y_{18p} \equiv 0 \bmod 17$

Hasse's Theorem

SENZA DIMOSTRAZIONE

UNIVERSITÀ DI PISA

- Hasse's theorem
  - Given an elliptic curve E modulo p, the number of points on the curve is denoted by #E and is bounded by:

$$p + 1 - 2\sqrt{p} \le \#E \le p + 1 + \sqrt{p}$$

CARDINALITA' CURVA → NUMERO DI PUNTI FINITA ANCHE SE NON SEMPLICE DA CALCOLARE

  - The number of points is roughly in the range of p (Hasse's bound)
  - Example – If you need an EC with 2^{160} points, you have to use a prime p of about 160 bit

To set up DL cryptosystems it is important to know the order of the group. Even though knowing the exact number of points on a curve is an elaborate task, we know the approximate number due to Hasse's theorem.

ECDLP – point multiplication

UNIVERSITÀ DI PISA

- Elliptic Curved Discrete Logarithm Problem (ECDLP)
  - Given is an elliptic curve E. We consider a primitive element P and another element T. The DL problem is finding the integer d, where 1 ≤ d ≤ #E, such that:

$$\underbrace{P + P + \ldots + P}_{d \text{ times}} = d \cdot P = T$$

FACILE => DATI d e P CALCOLARE T = d × P

  - d is the private key, T is the public key
  - Point multiplication $\stackrel{\text{def}}{=}$ $\boxed{T = d \cdot P}$ POINT MULTIPLICATION

DIFFICILE => DATI P AND T CALCOLARE $d = T/P$ LOGARITMO

In cryptosystems, d is the private key which is an integer, while the public key T is a point on the curve with coordinates T = (xT ,yT). The operation is called *point multiplication*, since we can formally write T = d·P.

Square-and-multiply

UNIVERSITÀ DI PISA

- Point multiplication is analogue to exponentiation in multiplicative groups $(\mathbb{Z}_p^*, \times)$
- We can adopt the square-and-multiply algorithm
- Example
  - $26P = (11010)_2P = (d_4d_3d_2d_1d_0)2\ P$
  - Step
    - #0 P = **1**P — init setting, bit processed: $d_4 = 1$
    - #1a P+P = 2P = **10**P — DOUBLE, bit processed: d_3
    - #1b 2P+P = 3P = 10P+1P = **11**P — ADD, since $d_3 = 1$
    - #2a 3P+3P = 6P = 2(11P) = **110**P — DOUBLE, bit processed: d_2
    - #2b — no ADD, since $d_2 = 0$
    - #3a 6P+6P = 12P = 2(110P) = **1100**P — DOUBLE, bit processed: d_1
    - #3b 12P+P = 13P = 1100P+1P = **1101**P — ADD, since $d_1 = 1$
    - #4a 13P+13P = 26P = 2(1101P) = **11010**P — DOUBLE, bit processed: d_0
    - #4b — no ADD, since $d_0 = 0$

EC Cryptosystem

- Private key: d
- Public key: T
- Geometrical interpretation of ECDLP
  - Given P, we compute 2P, 3P,…, d·P = T, we actually jump back and forth on the EC
  - Given the starting point P and the final point T (public key), the adversary has to figure out how often we “jumped” on the EC