Elliptic Curve Cryptosystems - ECC

CS 411/507 - Cryptography

Erkay Savaş & Atıl Utku Ay

Faculty of Engineering and Natural Sciences
Sabancı University

December 3, 2023

Overview

- Another public key cryptography algorithm (in addition to RSA).
- Elliptic curves as algebraic/geometric entities have been studied extensively for the past 150 years.
  - Studies revealed a rich and deep theory suitable to cryptographic usage.
- First proposed for cryptographic usage in 1985 independently by Neal Koblitz and Victor S. Miller

Security Level of ECC

- 160-bit key length is equivalent in cryptographic strength to 1024-bit RSA.
  - 313-bit ECC is equivalent to 4096-bit RSA

Algorithm family	**Bit length**
Integer Factorization (IF)	2048/3072/7680/15360
Discrete Logarithm (DL)	2048/3072/7680/15360
Elliptic Curve Discrete Logarithm (ECDL)	224/256/384/512
Block cipher	112/128/192/256
Hash Functions (SHA-2 & SHA-3)	224/256/384/512

Table: Security levels of PKCs

Overview

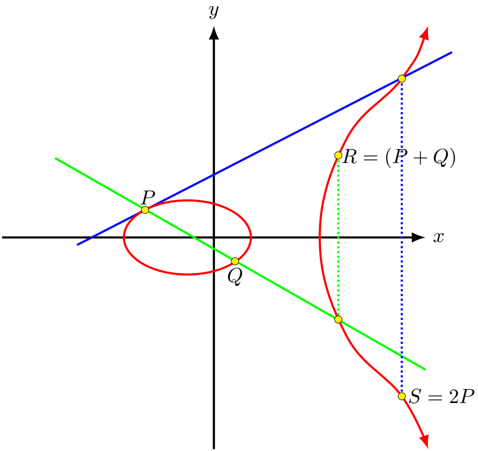
- Many cryptosystems require the use of algebraic groups.
  - Discrete logarithm as a hard problem
  - Elliptic curves may be used to form elliptic curve groups
  - Discrete logarithm problem in elliptic curve groups
  - Elliptic curves received their name from their relation to <u>elliptic integrals</u>

$$\int_{z_1}^{z_2} \frac{dx}{\sqrt{x^3 + ax + b}} \text{ and } \int_{z_1}^{z_2} \frac{x\,dx}{\sqrt{x^3 + ax + b}}$$

  - Used in the computation of the arc length of ellipses.

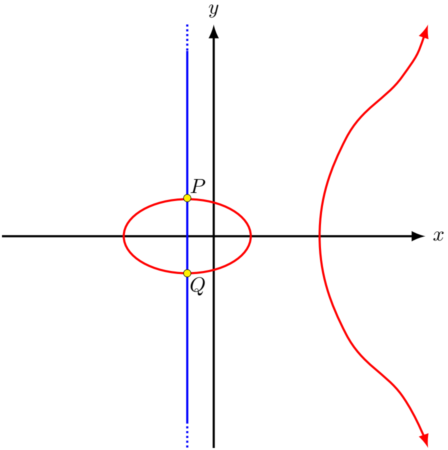
A Geometric Approach

- Elliptic curves over real numbers
  $y^2 = x^3 + ax + b$



A Geometric Approach

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

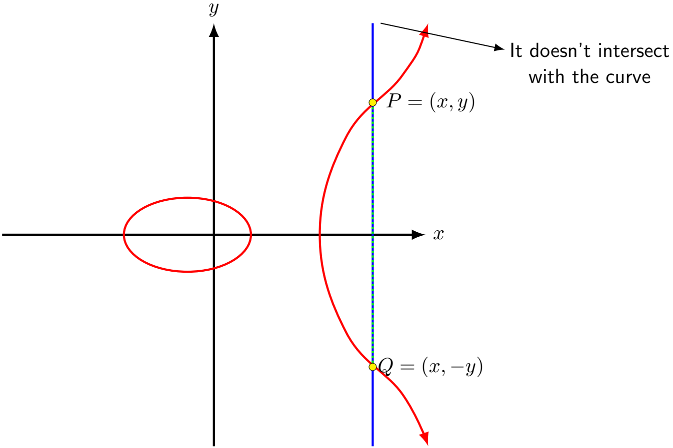


No intersection!?

<u>Abstraction</u>:
we say the line intersects with the curve at infinity.
<u>Definition</u>:
the intersection point is called "point at infinity" and denoted as $\mathcal{O}$.

Additive Inverse

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



- $Q + P = \mathcal{O} \rightarrow Q = -P => -(x, y) = (x, -y)$

Addition Law 1/3

- The elliptic curve equation
  - $E: y^2 = x^3 + ax + b$
- Two points on E:
  - $P = (x_p, y_p)$ and $Q = (x_q, y_q)$
- Addition
  - $R = P + Q = (x_r, y_r)$.
- The line L going through P and Q can be written as
  - $y = \lambda x + \beta$

    where $\lambda = (y_q - y_p)/(x_q - x_p)$ (the slope when $P \neq Q$)

    $\beta = y_p - \lambda x_p$, then

    $y = \lambda(x - x_p) + y_p$

Addition Law 2/3

- If $P \neq Q$,
  - $(\lambda x + \beta)^2 = x^3 + ax + b$
  - $x^3 - \lambda^2 x^2 + (a - 2\lambda\beta)x + b - \beta^2 = 0$
- We know
  - $(x - x_p)(x - x_q)(x - x_r) = 0$
  - $x^3 - (x_p + x_q + x_r)x^2 + (x_p x_q + x_p x_r + x_q x_r)x - x_p x_q x_r = 0$
- Therefore,
  - $\lambda^2 = x_p + x_q + x_r$
  - $x_r = \lambda^2 - (x_p + x_q)$
  - $-y_r = \lambda(x_r - x_p) + y_p$
  - $y_r = \lambda(x_p - x_r) - y_p$

Addition Law 3/3

- If $P = Q$, slope of the tangent can be calculated
  - by differentiating the curve equation at $P = (x_p, y_p)$
  - $y^2 = x^3 + ax + b$
  - $2y_p y' = 3{x_p}^2 + a \rightarrow \lambda = (3{x_p}^2 + a)/2y_p$
- Thus the tangent line is
  - $y = \lambda(x - x_p) + y_p$
- The point $R = 2P = (x_r, y_r)$
  - $(x - x_p)^2(x - x_r) = 0$
  - $x^3 - (x_r + 2x_p)x^2 + (2x_p x_r + {x_p}^2)x - {x_p}^2 x_r = 0$
  - $x_r = \lambda^2 - 2x_p$
  - $y_r = \lambda(x_p - x_r) - y_p$

Discriminant of a Curve

- We are interested in curves that are non-singular.
- Geometrically, this means that the curve has no self-intersections, cusps, or isolated points.
- To guarantee this for the discriminant Δ, we should have
- $\Delta = -16(4a^3 + 27b^2) \neq 0$

Elliptic Curves over $GF(p)$

- Solutions to
  - $y^2 \equiv x^3 + ax + b \bmod p$, where $0 \leq a, b < p$ forms the elliptic curve group.
  - Each solution is called a point on the curve.
- Two Points:
  - $P = (x_p, y_p)$ and $Q = (x_q, y_q)$ $\; 0 \leq x_p, y_p, x_q, y_q < p$
- Point Addition Rule:
  - $R = (x_r, y_r) = P + Q$
  - If $P \neq Q \rightarrow \lambda \equiv (y_p - y_q)/(x_p - x_q) \bmod p$
  - If $P = Q \rightarrow \lambda \equiv (3{x_p}^2 + a)/2y_p \bmod p$
  - $x_r \equiv \lambda^2 - x_p - x_q \bmod p$
  - $y_r \equiv -y_p + \lambda(x_p - x_r) \bmod p$

Example - 1

- $E : y^2 \equiv x^3 + 2x + 1 \bmod 5$
  - $\Delta = -16(4a^3 + 27b^2) = -16(4 \times 2^3 + 27) \bmod 5 \equiv -1(4) \equiv 1$
  - The points on E are the pairs $(x, y) \bmod 5$ that satisfies the equation, along with the point at infinity.
- The possibilities for x are $GF(5) = \{0, 1, 2, 3, 4\}$
  - $x = 0 \rightarrow y^2 \equiv 1 \bmod 5 \rightarrow y \equiv$
  - $x = 1 \rightarrow y^2 \equiv 4 \bmod 5 \rightarrow y \equiv$
  - $x = 2 \rightarrow y^2 \equiv 3 \bmod 5 \rightarrow y \equiv$
  - $x = 3 \rightarrow y^2 \equiv 4 \bmod 5 \rightarrow y \equiv$
  - $x = 4 \rightarrow y^2 \equiv 3 \bmod 5 \rightarrow y \equiv$
- Therefore the points are
  - $(0, 1)$, $(0, 4)$, $(1, 2)$, $(1, 3)$, $(3, 2)$, $(3, 3)$, $(0, 0)$

Example - 1

- Let us compute $(1, 3) + (3, 2)$.
- The slope
  - $\lambda \equiv 2$
- The first coordinate
  - $x_r \equiv \lambda^2 - x_p - x_q \mod p$
  - $x_r \equiv 4 - 1 - 3 = 0$
  - $y_r \equiv -y_p + \lambda(x_p - x_r) \mod p$
  - $y_r \equiv -3 + 2(1 - 0) = -1 \equiv 4$
- The resulting point
  - $(1, 3) + (3, 2) = (0, 4)$
  - is also on the curve (closed)

Example - 1

- Let us take $P = (1, 3)$
- Compute $P + P = 2P = (3, 2)$
- $P + 2P = 3P = (0, 4)$
- $P + 3P = 4P = (0, 1)$
- $P + 4P = 5P = (3, 3)$
- $P + 5P = 6P = (1, 2)$
- $P + 6P = 7P = (0, 0)$
- $P + 7P = 8P = (1, 3)$

Example - 1

$+$	$(0,0)$	$(1,3)$	$(3,2)$	$(0,4)$	$(0,1)$	$(3,3)$	$(1,2)$
$(0,0)$	$(0,0)$	$(1,3)$	$(3,2)$	$(0,4)$	$(0,1)$	$(3,3)$	$(1,2)$
$(1,3)$	$(1,3)$	$(3,2)$	$(0,4)$	$(0,1)$	$(3,3)$	$(1,2)$	$(0,0)$
$(3,2)$	$(3,2)$	$(0,4)$	$(0,1)$	$(3,3)$	$(1,2)$	$(0,0)$	$(1,3)$
$(0,4)$	$(0,4)$	$(0,1)$	$(3,3)$	$(1,2)$	$(0,0)$	$(1,3)$	$(3,2)$
$(0,1)$	$(0,1)$	$(3,3)$	$(1,2)$	$(0,0)$	$(1,3)$	$(3,2)$	$(0,4)$
$(3,3)$	$(3,3)$	$(1,2)$	$(0,0)$	$(1,3)$	$(3,2)$	$(0,4)$	$(0,1)$
$(1,2)$	$(1,2)$	$(0,0)$	$(1,3)$	$(3,2)$	$(0,4)$	$(0,1)$	$(3,3)$

Example - 2

- $E : y^2 \equiv x^3 + x + 3 \bmod 7$
  - Points: $(4, 1)$, $(4, 6)$, $(5, 0)$, $(6, 1)$, $(6, 6)$, $(0, 0)$
  - Group order: 6
- $P = (5, 0)$
  - $2P = (0, 0)$, $3P = (5, 0)$
- $Q = (4, 1)$
  - $2Q = (6, 6)$, $3Q = (5, 0)$, $4Q = (6, 1)$, $5Q = (4, 6)$, $6Q = (0, 0)$
- $S = (6, 1)$
  - $2S = (6, 6)$, $3S = (0, 0)$, $4S = (6, 1)$

Number of Points on Curve

- Generally, it is not easy to count the points on a curve.
- Assume that the underlying field K (the field over which the elliptic curve is constructed) has p elements
- Then for the number of points n on the curve E defined over K, we can write
$$\left|n - p - 1\right| < 2\sqrt{p}$$
- Hasse bound (1930s)

Example: Number of Points on Curve

- Previous example:
  - $E : y^2 \equiv x^3 + 2x + 1 \bmod 5$
- The points on E are the pairs $(x, y) \bmod 5$ that satisfies the equation, along with the point at infinity.
- The points are
  - $(0,1)$, $(0,4)$, $(1,2)$, $(1,3)$, $(3,2)$, $(3,3)$ and $\mathcal{O} = (0,0)$.
- Therefore, $\#E(F_5) = n = 7$

$$|n - p - 1| < 2\sqrt{p} \rightarrow |7 - 5 - 1| = 1 < 4.472$$

Elliptic Curve DL Problem

- Scalar Multiplication:
- $Q = kP$, where P and Q are points, k is an integer

$$kP = \underbrace{P + P + \cdots + P}_{k \text{ times}}$$

- Scalar multiplication is repeated point addition
  - However, it is not efficient (or more precisely, computationally infeasible) to use the repeated addition method.

Example: $Q = 53P$
$Q = \mathcal{O}$
$Q = 2\mathcal{O} + P = P$
$Q = 2P + P = 3P$
$Q = 6P$

$53 = (110101)_2$
$Q = 12P + P = 13P$
$Q = 26P$
$Q = 52P + P$

Binary Left-to-Right Algorithm

Algorithm 1 Binary Left-to-Right Algorithm

Input: P a point on the curve and $k \geq 1$ an integer $(k = k_{k-1}, \ldots, k_1, k_0)$
Output: $Q \equiv kP$

```
Q := O
for i = k − 1 downto 0 do
    Q := Q + Q
    if k_i = 1 then
        Q := Q + P
    end if
end for
return Q
```

Binary Left-to-Right Algorithm - Example

- $Q = 53P$
  - $53 = (110101)_2$
- Step 0: $Q = \mathcal{O}$
- Step 1: $53 = (1.....)_2$
  - $Q = 2\mathcal{O} + P = P$
- Step 2: $53 = (11....)_2$
  - $Q = 2P + P = 3P$
- Step 3: $53 = (110...)_2$
  - $Q = 6P$
- Step 4: $53 = (1101..)_2$
  - $Q = 12P + P = 13P$
- Step 5: $53 = (11010.)_2$
  - $Q = 26P$
- Step 6: $53 = (110101)_2$
  - $Q = 52P + P$

Algorithm 2 Binary Right-to-Left Algorithm

Input: P a point on the curve and $k \geq 1$ an integer

Output: $Q \equiv kP$

```
Q := O; T := P
while k ≠ 0 do
    if k is odd then
        Q := Q + T
    end if
    k := k >> 1
    if k ≠ 0 then
        T := 2T
    end if
end while
return Q
```

EC/DL Problem

- Definition:
  - Given points P and Q in the group, find a number k such that $Q = kP$
  - VERY HARD PROBLEM !
- In elliptic curve schemes, the most time consuming operation is scalar multiplication.
- The security of the elliptic curve cryptosystems depends on the size of k.
- In real applications k is large.
- The minimum bit length of k is 256 for commercial applications.

Elliptic Curve Cryptosystems

- Discrete logarithm problem (DLP) over elliptic curves is harder than the DLP over integers $\bmod p$.
- The most efficient method for computing DL, which is the "index calculus method", seems to have no counterpart for elliptic curves.
- Therefore, it is possible to use much smaller primes or finite fields with elliptic curves to achieve the same level of security.

- The complexity of discrete logarithm algorithms
  1. Index-calculus method:
     - Minimum security requirement in $\mathbb{Z}_p^*$: $(p-1) > 2^{2048}$
  2. Shanks's algorithm (baby-step giant-step)
     - Complexity $(n)^{\frac{1}{2}}$
     - Minimum security requirement: $(n) > 2^{224}$
  3. Pohlig-Hellman algorithm:
     - $n = p_1 p_2 p_3 \ldots p_j \rightarrow$ complexity $O((p_j)^{\frac{1}{2}})$
     - Minimum security requirement: $(n) > 2^{224}$
- This is why 224-bit ECDL is equivalent in cryptographic strength to 2048-bit DL.

Elliptic Curve Cryptosystems

- It is easy to change classical systems based on DL into one using elliptic curves:
  1. Change modular multiplication to elliptic curve point addition.
  2. Change modular exponentiation to multiplying an elliptic curve point by an integer (scalar point multiplication).

ECDH Key Exchange

- $E : y^2 \equiv x^3 + ax + b \mod p$.
- Base point P on an elliptic curve. $ord(P) = n$

Alice

1. Picks a random s_A
   $2 \leq s_A < n - 1$
2. Computes $Q_A = s_A P$
3. Publishes Q_A
4. Computes k_{AB}
   $k_{AB} = s_A Q_B$
   $k_{AB} = s_A s_B P$

Bob

1. Picks a random s_B
   $2 \leq s_B < n - 1$
2. Computes $Q_B = s_B P$
3. Publishes Q_B
4. Computes k_{BA}
   $k_{BA} = s_B Q_A$
   $k_{BA} = s_B s_A P$

Session key: $k = k_{AB} = k_{BA} = s_A s_B P$

Example: ECDH Key Exchange

- $E : y^2 \equiv x^3 + x + 7206 \bmod 7211$
- a base point $P = (3, 5)$.
- $s_A = 12$
- $s_B = 23$
- $Q_A = s_A P = 12 \times (3, 5) = (1794, 6375)$.
- $Q_B = s_B P = 23 \times (3, 5) = (3861, 1242)$
- $s_A Q_B = 12 \times (3861, 1242) = (1472, 2098)$.
- $s_B Q_A = 23 \times (1794, 6375) = (1472, 2098)$.

DSA - Signature Scheme

- Domain parameters: (q, p, g)
- Key pair: $0 < s_A < q$ and $\beta = g^{s_A} \bmod p$
- Signature generation:
  - $h = H(m)$
  - $r = (g^k \bmod p) \bmod q$ and $s = k^{-1}(h + s_A r) \bmod q$.
- Signature verification
  - $h = H(m)$, $u_1 = s^{-1} h \bmod q$ and $u_2 = s^{-1} r \bmod q$.
  - $v = (g^{u_1} \beta^{u_2} \bmod p) \bmod q$.
  - Bob accepts the signature if and only if $v = r$.

EC Digital Signature Algorithm

- ECDSA
  - Alice wants to sign a message m.
  - Domain parameters
    - An elliptic curve E over $GF(p)$ and a base point P on the curve
    - base point P is a generator.
    - The number of points on E, n is known and assume n also a prime integer
  - She chooses a secret integer $s_A < n - 1$ and computes $Q_A = s_A P$.
  - Curve parameters (i.e. a, b, p, and P) and her public key Q_A are published
  - s_A is kept private.

ECDSA

- Signing the message m:
  1. She computes $h = HASH(m)$
  2. She selects a random integer k such that $0 < k < n$.
  3. computes $R = kP = (x_r, y_r)$
  4. $r = x_r \bmod n$
  5. $s = k^{-1}(h + s_A r) \bmod n$.
     - Alice's signature for m is (r, s).
- Verifying the signature given m and Q_A and curve parameters
  1. Bob computes $h = HASH(m)$
  2. $u_1 = s^{-1}h \bmod n$ and $u_2 = s^{-1}r \bmod n$
  3. $V = u_1P + u_2Q_A = (x_v, y_v)$
  4. Accepts if $x_v \equiv r \bmod n$.