Elliptic Curve Cryptography

Outline

- [1] Elliptic Curves over R
- [2] Elliptic Curves over GF(p)
- [3] Properties of Elliptic Curves
- [4] Computing Point Multiples on Elliptic Curves
- [5] ECDLP
- [6] ECDSA

[1] Elliptic curves over R

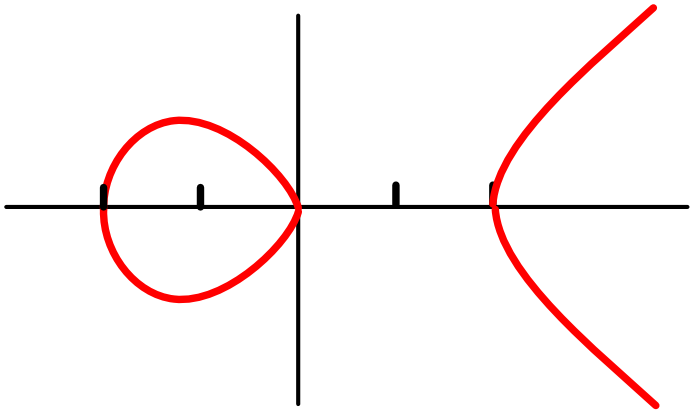
- Definition

  Let $a, b \in \boldsymbol{R},\ 4a^3 + 27b^2 \neq 0$

$$E = \left\{ (x, y) \in \boldsymbol{R} \times \boldsymbol{R} \middle| y^2 = x^3 + ax + b \right\} \cup \{ \mathrm{O} \}$$

- Example:

$$E : y^2 = x^3 - 4x$$



- Group operation +

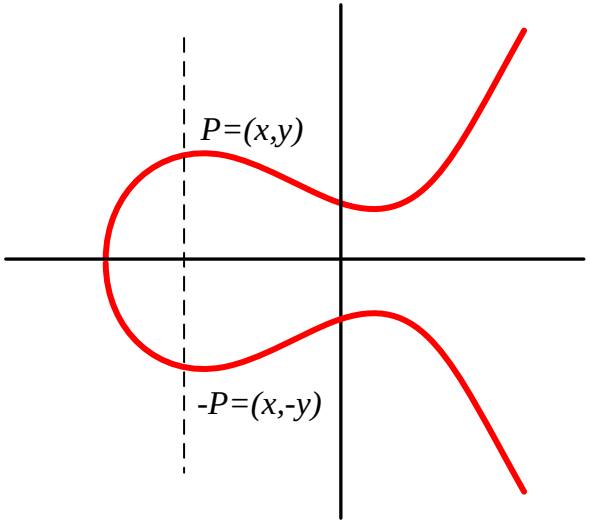
  The point of infinity, O, will be the identity element

  Given $P, Q \in E, P = (x_1, y_1), Q = (x_2, y_2)$

$$P + \mathrm{O} = \mathrm{O} + P$$

If $x_1 = x_2$, and $y_1 = -y_2$, then $P + Q = \mathrm{O}$

(i.e. $-P = -(x_1, y_1) = (x_1, -y_1)$)



- Group operation +

  Given $P, Q \in E, P = (x_1, y_1), Q = (x_2, y_2)$

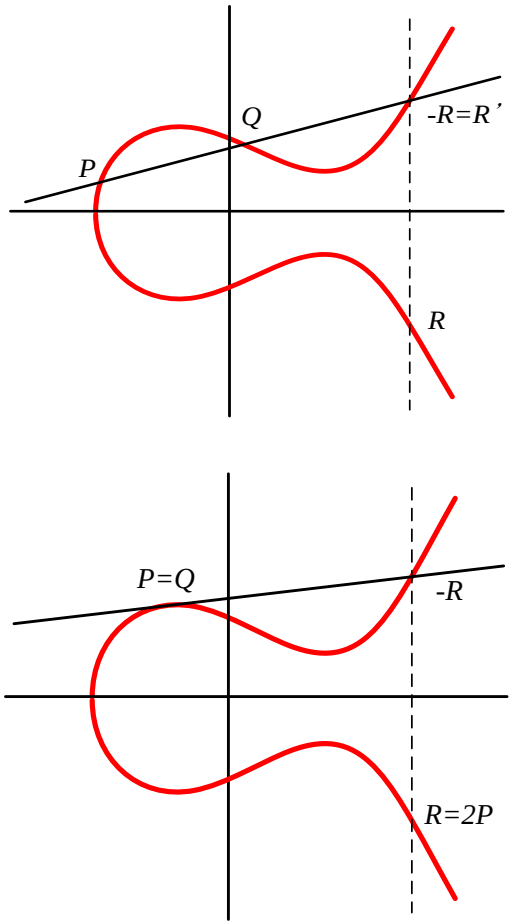
  Compute $R = P + Q = (x_3, y_3)$

  - Addition $(P \neq Q)$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$
$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = (x_1 - x_3)\lambda - y_1$$

  - Doubling $(P = Q)$

$$\lambda = \frac{3x_1^2 + a}{2y_1}$$
$$x_3 = \lambda^2 - 2x_1$$
$$y_3 = (x_1 - x_3)\lambda - y_1$$



- Example (addition):
  Given $E: y^2 = x^3 - 25x$
  - $P = (x_1, y_1) = (0,0),\ Q = (x_2, y_2) = (-5,0),\ P + Q = (x_3, y_3)$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - 0}{-5 - 0} = 0$$

$$x_3 = \lambda^2 - x_1 - x_2 = 0^2 - 0 - (-5) = 5$$

$$y_3 = (x_1 - x_3)\lambda - y_1 = (0 - 5) \times 0 - 0 = 0$$

- Example (doubling):
  Given $E: y^2 = x^3 - 25x$
  - $P = (x_1, y_1) = (-4, 6),\ 2P = (x_2, y_2)$

$$\lambda = \frac{3x_1^{\,2} + a}{2y_1} = \frac{3(-4)^2 - 25}{2 \times 6} = \frac{23}{12}$$

$$x_2 = \lambda^2 - 2x_1 = \left(\frac{23}{12}\right)^2 - 2 \times (-4) = \frac{1681}{144}$$

$$y_2 = (x_1 - x_2)\lambda - y_1 = \left(-4 - \frac{1681}{144}\right) \times \frac{23}{12} - 6 = -\frac{62279}{1728}$$

[2] Elliptic Curves over GF(p)

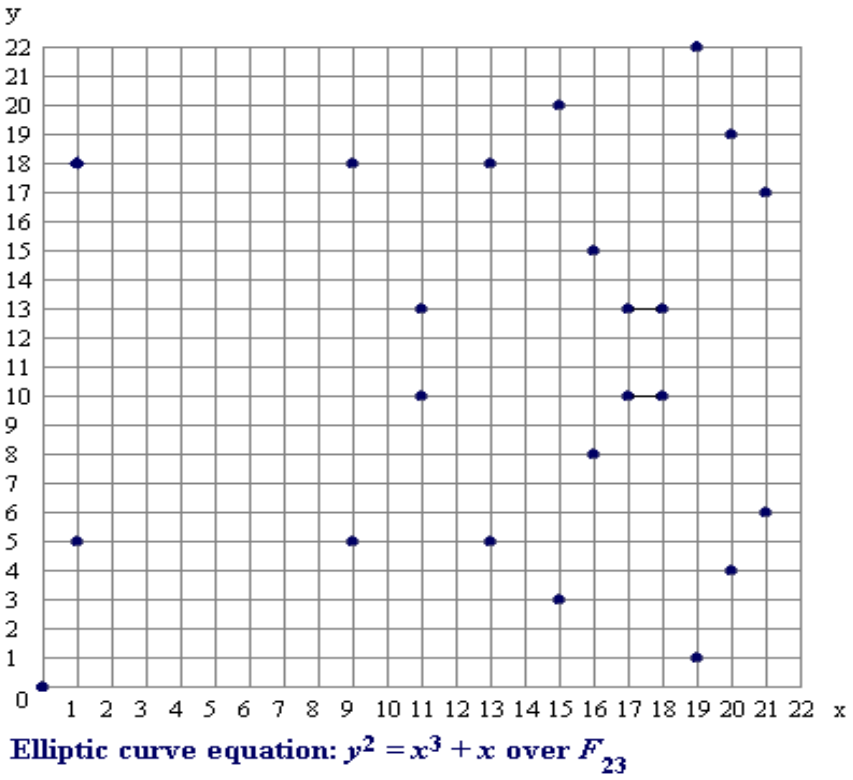
- Definition

  Let $p > 3, a, b \in Z_p,\ 4a^3 + 27b^2 \neq 0 \pmod{p}$

$$E = \left\{ (x, y) \in Z_p \times Z_p \,\middle|\, y^2 \equiv x^3 + ax + b \pmod{p} \right\} \cup \{ \mathrm{O} \}$$

- Example:

$$E : y^2 = x^3 + x \text{ over } Z_{23}$$



Elliptic curve equation: $y^2 = x^3 + x$ over F_{23}

- Example:

$E : y^2 = x^3 + x + 6$ over Z_{11}

Find all *(x, y)* and O:

- Fix *x* and determine *y*
- O is an artificial point

12 *(x, y)* pairs plus O,
and have *#E=13*

x	$x^3 + x + 6$	quad res?	y
0	6	*no*	
1	8	*no*	
2	5	*yes*	4,7
3	3	*yes*	5,6
4	8	*no*	
5	4	*yes*	2,9
6	8	*no*	
7	4	*yes*	2,9
8	9	*yes*	3,8
9	7	*no*	
10	4	*yes*	2,9

- Example (continue):
  There are 13 points on the group $E(Z_{11})$ and so any non-identity point (i.e. not the point at infinity, noted as O) is a generator of $E(Z_{11})$.

  Choose generator $\alpha = (2,7)$
  Compute $2\alpha = (x_2, y_2)$

$$\lambda = \frac{3x_1^{\,2} + a}{2y_1} = \frac{3(2)^2 + 1}{2 \times 7} = \frac{13}{14} = 2 \times 3^{-1} = 2 \times 4 = 8 \mod 11$$

$$x_2 = \lambda^2 - 2x_1 = (8)^2 - 2 \times (2) = 5 \mod 11$$

$$y_2 = (x_1 - x_2)\lambda - y_1 = (2-5) \times 8 - 7 = 2 \mod 11$$

- Example (continue):
  Compute $3\alpha = (x_3, y_3)$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-7}{5-2} = 2 \mod 11$$

$$x_3 = \lambda^2 - x_1 - x_2 = 2^2 - 2 - 5 = 8 \mod 11$$

$$y_3 = (x_1 - x_3)\lambda - y_1 = (2-8) \times 2 - 7 = 3 \mod 11$$

So, we can compute

$$\alpha = (2,7) \quad 2\alpha = (5,2) \quad 3\alpha = (8,3)$$

$$4\alpha = (10,2) \quad 5\alpha = (3,6) \quad 6\alpha = (7,9)$$

$$7\alpha = (7,2) \quad 8\alpha = (3,5) \quad 9\alpha = (10,9)$$

$$10\alpha = (8,8) \quad 11\alpha = (5,9) \quad 12\alpha = (2,4)$$

- Example (continue):
  Let's modify ElGamal encryption by using the elliptic curve $E(Z_{11})$.
  Suppose that $\alpha = (2,7)$ and Bob's private key is 7, so

$$\beta = 7\alpha = (7,2)$$

  Thus the encryption operation is

$$e_K(x,k) = \big(k(2,7), x + k(7,2)\big),$$

  where $x \in E$ and $0 \le k \le 12$, and the decryption operation is

$$d_K(y_1, y_2) = y_2 - 7y_1.$$

- Example (continue):
  Suppose that Alice wishes to encrypt the plaintext x = (10,9) (which is a point on E).
  If she chooses the random value k = 3, then

$$y_1 = 3(2,7) = (8,3) \text{ and}$$
$$y_2 = (10,9) + 3(7,2) = (10,9) + (3,5) = (10,2)$$

  Hence $y = \big((8,3),(10,2)\big)$. Now, if Bob receives the ciphertext y, he decrypts it as follows:

$$x = (10,2) - 7(8,3) = (10,2) - (3,5)$$
$$= (10,2) + (3,6) = (10,9)$$

[3] Properties of Elliptic Curves

- Over a finite field Z_p, the order of $E(Z_p)$ is denoted by $\#E(Z_p)$.
- Hasse's theorem

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

- Group structure of $E(Z_p)$

  Let E be an elliptic curve defined over Z_p, and $p>3$. Then there exists positive integers n_1 and n_2 such that

$$(E,+) \cong Z_{n_1} \times Z_{n_2}$$

  Further,

$$n_2 | n_1 \text{ and } n_2 | (p-1)$$

[4] Computing Point Multiples on Elliptic Curves

- Use Double-and-Add
  (similar to square-and-multiply)
  Algorithm:
  DOUBLE-AND-ADD $(P,(c_{l-1},\ldots,c_0)),\ c_i \in \{0,1\}$

$$Q \leftarrow \mathrm{O}$$

$$\text{for } i \leftarrow l-1 \text{ downto } 0$$

$$\text{do} \begin{cases} Q \leftarrow 2Q \\ \text{if } c_i = 1 \\ \quad \text{then } Q \leftarrow Q + P \end{cases}$$

$$\text{return } (Q)$$

- Example:
  Compute 3895P

$$3895P = \underbrace{P + P + \cdots + P}_{3894\ \text{additions needed}}$$

$$= (111100110111)_2 P$$

$$= 2(2(2(2(2(2(2(2(2(2P + P) + P) + P))) + P) + P)) + P) + P) + P$$

$$\rightarrow 11\ \text{doublings and}\ 8\ \text{additions needed}$$

- Use Double-and-(Add or Subtract)
  - Elliptic curve has the property that additive inverses are very easy to compute.
  - Signed binary representation
    - Example:

$$11 = 8 + 2 + 1 = 16 - 4 - 1,$$

so

$$(c_4, c_3, c_2, c_1, c_0) = (0, 1, 0, 1, 1) \text{ or } (1, 0, -1, 0, -1)$$

are both signed binary representation of 11.

- Non-adjacent form (NAF)
  A signed binary representation $(c_{l-1}, \ldots, c_0)$ of an integer c is said to be in non-adjacent form provided that no two consecutive ci's are non-zero.

  - The NAF representation of an integer is unique.
  - A NAF representation contains more zeros than the traditional binary representation of a positive integer.

- Transform a binary representation of a positive integer *c* into a NAF representation
  Example:

$$
\begin{array}{cccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
 & & & & & \downarrow & & & & & & \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & -1 \\
 & & & & & \downarrow & & & & & & \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 \\
 & & & & & \downarrow & & & & & & \\
1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1
\end{array}
$$

  Hence the NAF representation of (1,1,1,1,0,0,1,1,0,1,1,1) is (1,0,0,0,-1,0,1,0,0,-1,0,0,-1)

- Double-and-(Add or Subtract)

Algorithm 6.5:

DOUBLE-AND-(ADD OR SUBTRACT) $(P,(c_{l-1},\ldots,c_0)),\ c_i \in \{0,\pm 1\}$

$Q \leftarrow \mathrm{O}$

for $i \leftarrow l-1$ downto 0

do
- $Q \leftarrow 2Q$
- if $c_i = 1$
  - then $Q \leftarrow Q + P$
- else if $c_i = -1$
  - then $Q \leftarrow Q - P$

return (Q)

- Example:
  Compute 3895P

$$3895P = \underbrace{P + P + \cdots + P}_{3894\ \text{additions needed}}$$

$$= (111100110111)_2 P$$

$$= 2(2(2(2(2(2(2(2(2(2P + P) + P) + P))) + P) + P)) + P) + P) + P$$

$$\rightarrow 11\ \text{doublings and}\ 8\ \text{additions needed}$$

$$= (1000(\text{-}1)0100(\text{-}1)00(\text{-}1))_2 P$$

$$= 2(2(2(2(2(2(2(2(2(2P))) - P)) + P))) - P))) - P$$

$$\rightarrow 12\ \text{doublings and}\ 4\ (\text{additions or subtractions})\ \text{needed}$$

[5] Elliptic Curve DLP

- Basic computation of ECC
  - Q = kP = $\underbrace{P+P+...+P}_{k \text{ times}}$
    where P is a curve point, k is an integer
- Strength of ECC
  - Given curve, the point P, and kP
    It is hard to recover k
    - Elliptic Curve Discrete Logarithm Problem (ECDLP)

Security of ECC versus RSA/ElGamal

- Elliptic curve cryptosystems give the most security per bit of any known public-key scheme.
- The ECDLP problem appears to be much more difficult than the integer factorisation problem and the discrete logarithm problem of Z_p. (no index calculus algo!)
- The strength of elliptic curve cryptosystems grows much faster with the key size increases than does the strength of RSA.

Elliptic Curve Security

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

NIST Recommended Key Sizes

ECC Benefits

ECC is particularly beneficial for application where:

- computational power is limited (wireless devices, PC cards)
- integrated circuit space is limited (wireless devices, PC cards)
- high speed is required.
- intensive use of signing, verifying or authenticating is required.
- signed messages are required to be stored or transmitted (especially for short messages).
- bandwidth is limited (wireless communications and some computer networks).

[6] Signature Scheme: ECDSA

- Digital Signature Algorithm (DSA)
  - Proposed in 1991
  - Was adopted as a standard on December 1, 1994
- Elliptic Curve DSA (ECDSA)
  - FIPS 186-2 in 2000

Digital Signature Algorithm (DSA)

L=0 mod 64,
512≤L≤1024

- Let p be a L-bit prime such that the DL problem in Z_p^* is intractable, and let q be a 160-bit prime that divides p-1. Let α be a q_{th} root of 1 modulo p.

  Define K={ (p,q,α,a,β): $\beta=\alpha^a \bmod p$ }

  p,q,α,β are the public key, a is private

- For a (secret) random number k, define

  sig (x,k)=(γ,δ), where

  $\gamma=(\alpha^k \bmod p) \bmod q$ and

  $\delta=(SHA\text{-}1(x)+a\gamma)k^{-1} \bmod q$

- For a message (x,(γ,δ)), verification is done by performing the following computations:

  $e_1=SHA\text{-}1(x)*\delta^{-1} \bmod q$

  $e_2=\gamma*\delta^{-1} \bmod q$

  ver(x,(γ,δ))=true iff. $(\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q=\gamma$

Elliptic Curve DSA

- Let p be a prime, and let E be an elliptic curve defined over F_p. Let A be a point on E having prime order q, such that DL problem in <A> is infeasible.

  Define *K*={ (p,q,E,A,m,B): B=mA }

  p,q,E,A,B are the public key, m is private

- For a (secret) random number k, define $sig_k(x,k)=(r,s)$,

  where $kA=(u,v)$, $r=u \bmod q$ and

  $s=k^{-1}(SHA\text{-}1(x)+mr) \bmod q$

- For a message (x,(r,s)), verification is done by performing the following computations:

  $i=SHA\text{-}1(x)*s^{-1} \bmod q$

  $j=r*s^{-1} \bmod q$

  $(u,v)=iA+jB$

  ver(x,(r,s))=true if and only if $u \bmod q=r$