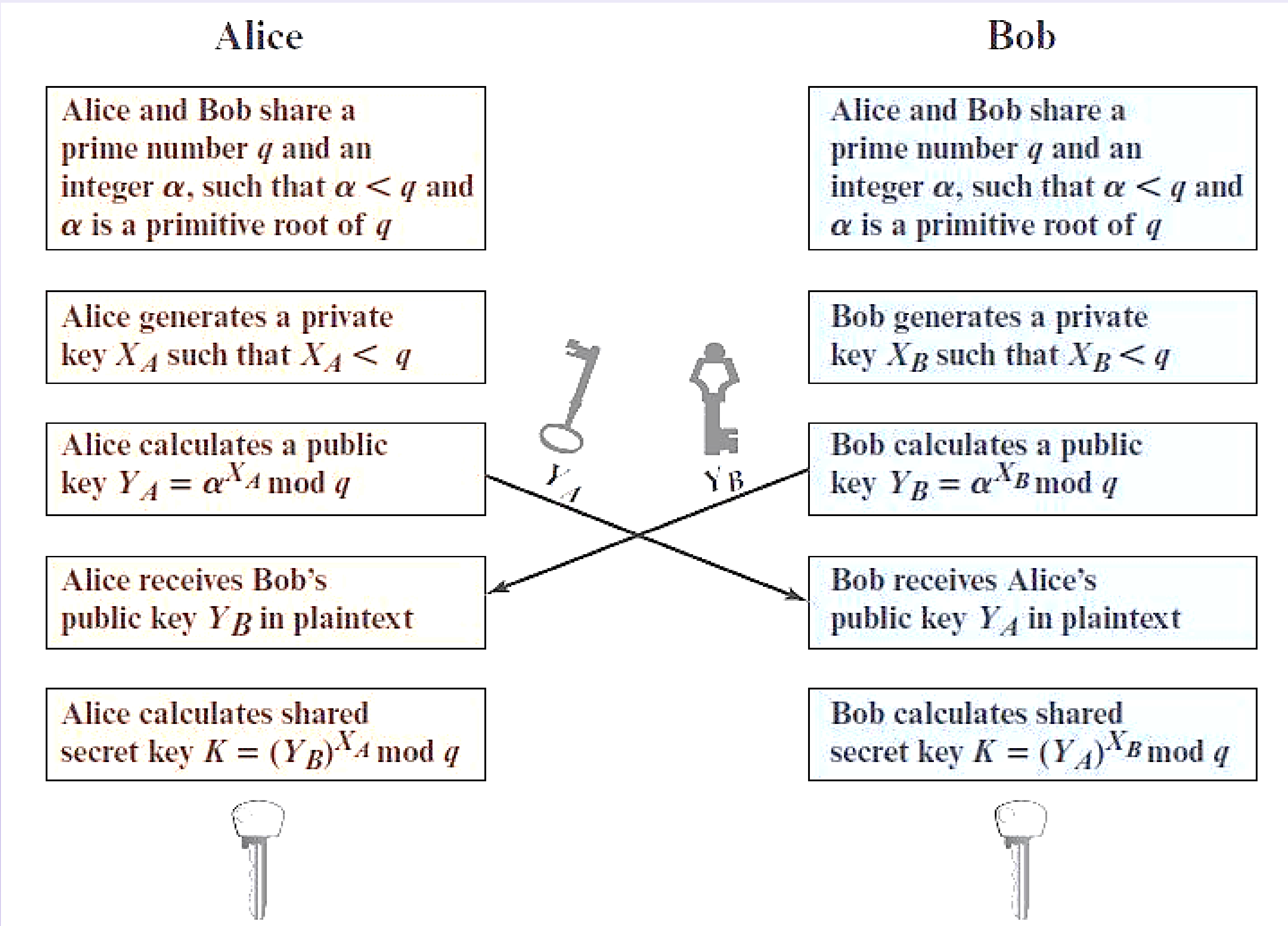
Chapter 10

Other Public-Key Cryptosystems

Diffie-Hellman Key Exchange

- First published public-key algorithm, 1976
- Purpose: enable two users to securely exchange a secret key over a public channel
- Operations are on group Z_q^*, where q is prime

DH key exchange protocol



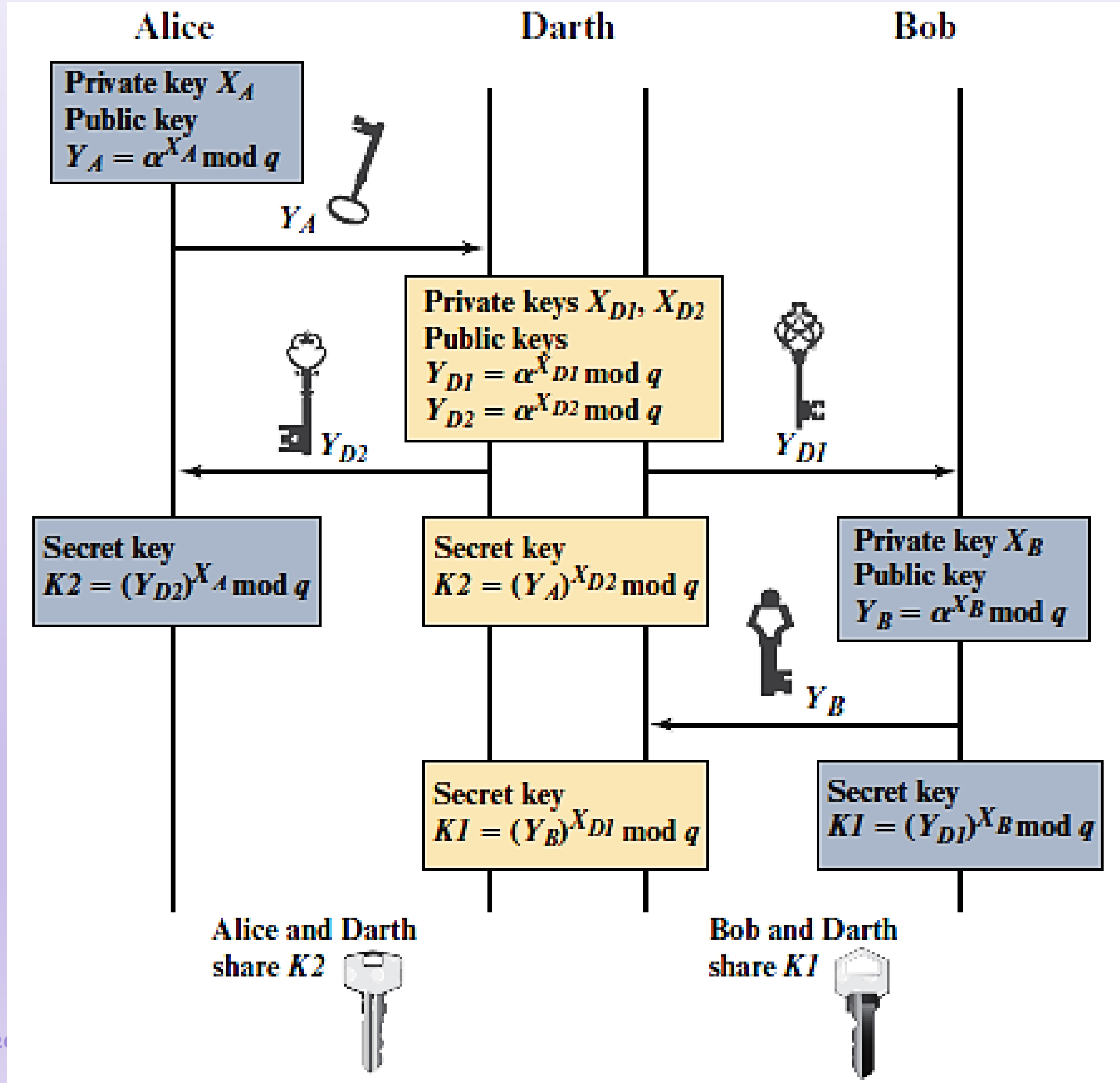
DH key exchange: example

- Global parameters: q=353, α=3
- Alice
  - choose X_A=97
  - compute Y_A=3^{97} mod 353=40
  - send Y_A to Bob
- Bob
  - choose X_B=233
  - compute Y_B=3^{233} mod 353=248
  - send Y_B to Alice
- Alice: compute K = $Y_B^{X_A}$ = 248^{97} mod 353 = 160
- Bob: compute K = $Y_A^{X_B}$ = 40^{233} mod 353 = 160

DH key exchange: security

- The DH problem
  - given $(\mathrm{q}, \alpha, Y_A, Y_B)$, compute $K = \alpha^{X_A X_B} \bmod q$, where $Y_A = \alpha^{X_A} \bmod \mathrm{q}$ and $Y_B = \alpha^{X_B} \bmod q$
- DH problem is no harder than dlog problem
  - Solving DL problem → solving DH problem
  - However the vice versa is not known yet
- Attack: man-in-the-middle attack

Man-in-the-middle attack



ElGamal cryptography

- Taher ElGamal, 1984
  - Public-key encryption
  - Digital signature (introduced later)
- Operations are on group Z_q^*, where q is prime

ElGamal encryption

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2K^{-1}) \bmod q$

ElGamal encryption: example

- Global parameter: q=19, α=10
- Alice's key generation:
  - Choose X_A=5, compute Y_A=10^5 mod 19=3
  - PU_A =(q, α , Y_A)= (19, 10, 3), PR_A =(q, α, X_A)= (19, 10, 5)
- Encryption: M=17, PU_A= (19, 10, 3)
  - Pick k=6, compute C=(10^6 mod 19, 17x3^6 mod 19)=(11, 5)
- Decryption: C=(11, 5),PR_A=(19, 10, 5)
  - Compute M = 5/(11^5 mod 19) mod 19
    = 5/7 mod 19 = 5x11 mod 19 = 17

ElGamal encryption: security

- compute private key → solve dlog problem
  - Given (q, α, Y_A) , compute $X_A = \mathrm{dlog}_{\alpha,\mathrm{q}} Y_A$
- compute plaintext → solve the DH problem
  - Given $(\mathrm{q}, \alpha, Y_A, C_1, C_2)$, compute $M = C_2/\alpha^{kX_A} \bmod q$, where $Y_A = \alpha^{X_A} \bmod q,\ C_1 = \alpha^k \bmod q$ and $C_2 = M\alpha^{kX_A} \bmod q$
  - When M is solved, $K = \alpha^{kX_A} \bmod q = C_2/M \bmod q$
- k is used only once. Otherwise,
  - Two ciphertexts
    - $\left(C_{1,1}, C_{2,1}\right) = (\alpha^k\ mod\ q, M_1\alpha^{kX_A}\ mod\ q)$
    - $\left(C_{1,2}, C_{2,2}\right) = (\alpha^k\ mod\ q, M_2\alpha^{kX_A}\ mod\ q)$
  - $C_{2,2}/C_{2,1} \bmod q = M_2/M_1 \bmod q$
  - If M_1 is known, M_2 is compromised

ElGamal encryption: computation

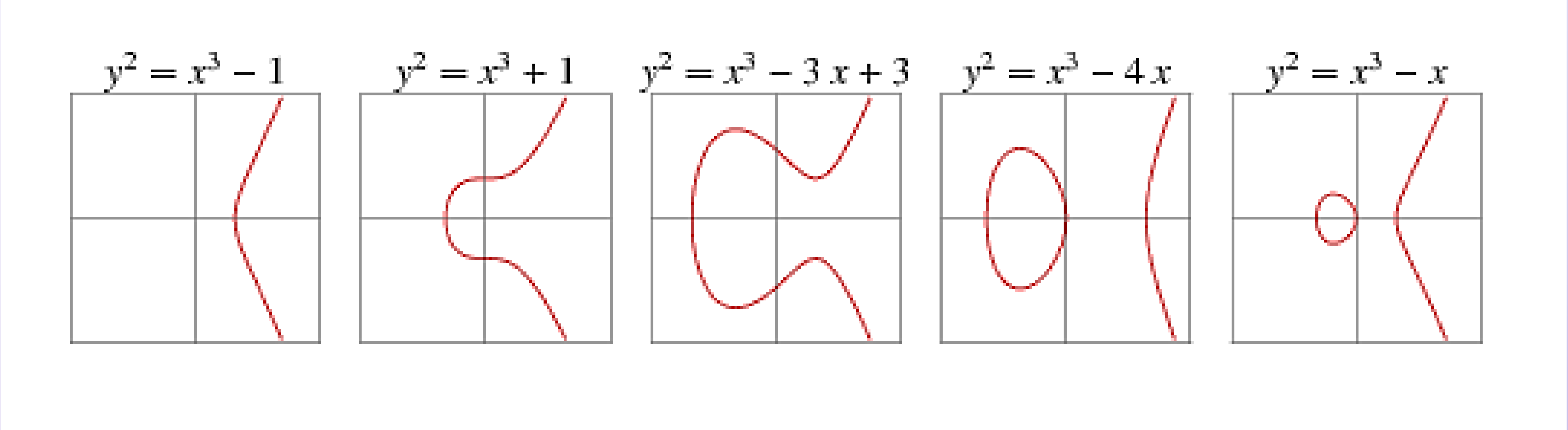
- Two modular exponentiations for an encryption
- Ciphertext expansion
  - $|C| = |C_1| + |C_2| = 2|M|$

Key length problem

- RSA and ElGamal encryption
  - The key length has increased over years because of security concern
  - RSA modulus n
    - 1024 bits, 2002
    - 2048 bits, 2015
  - ElGamal cryptosystem
    - modulus q: 2048 bits, 2017
    - private key X_A: 160-240 bits
- Elliptic curve cryptography (ECC)
  - IEEE P1363 Standard for Public-Key Cryptography
  - Shorter key length: 256 bits
  - fast encryption/decryption
  - Suitable for mobile devices, such as, IoT

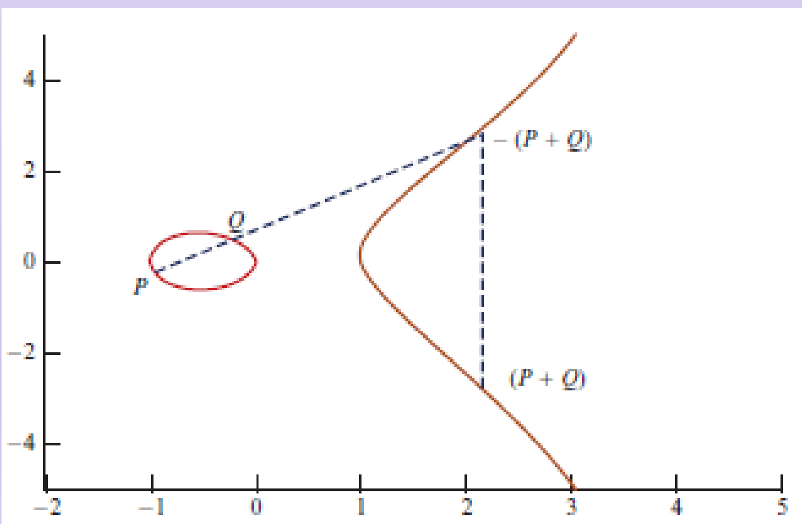
Elliptic Curve over reals

- Weierstrass equation: E: $y^2 = x^3 + ax + b$
  - $4a^3 + 27b^2 \neq 0$: non-singular
- Examples

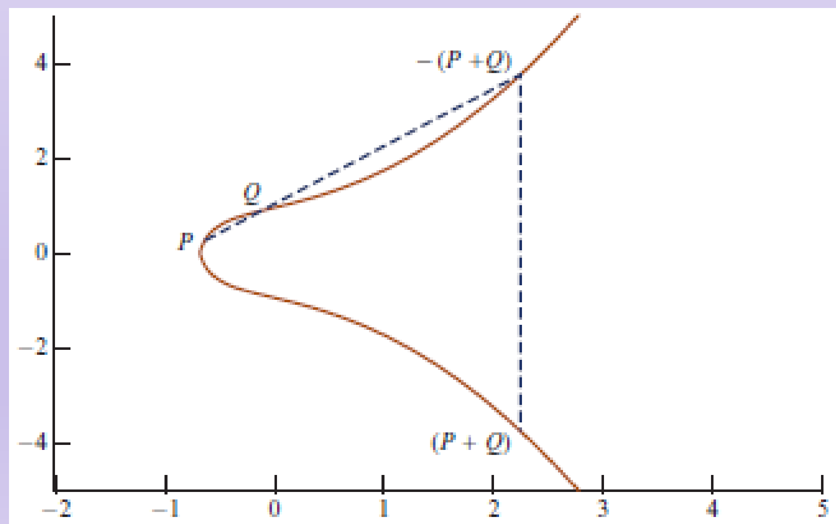


Additive group on elliptic curves

- Two ingredients
  - The infinity point: O
  - The sum of three points on a line is O
- group elements: all points in the curve and identity O
- addition and inverse: illustrated on graphs



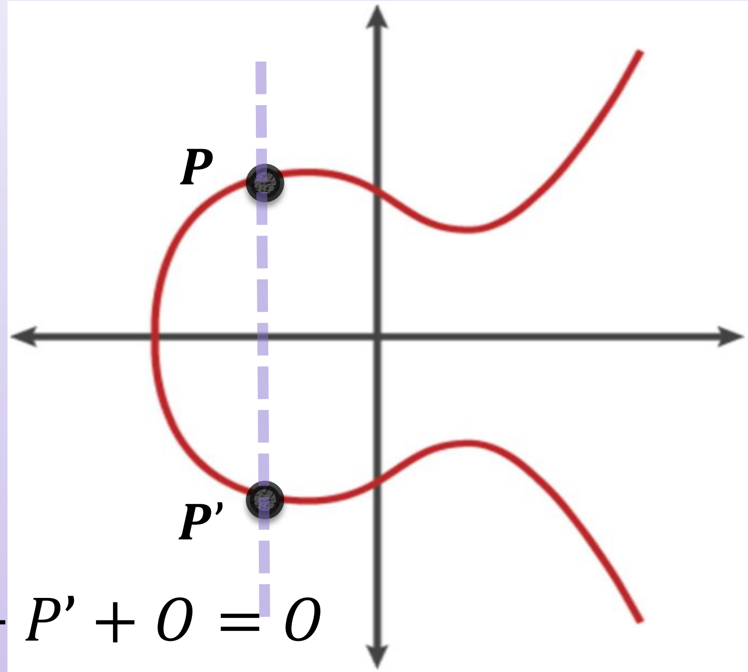
(a) $y^2 = x^3 - x$



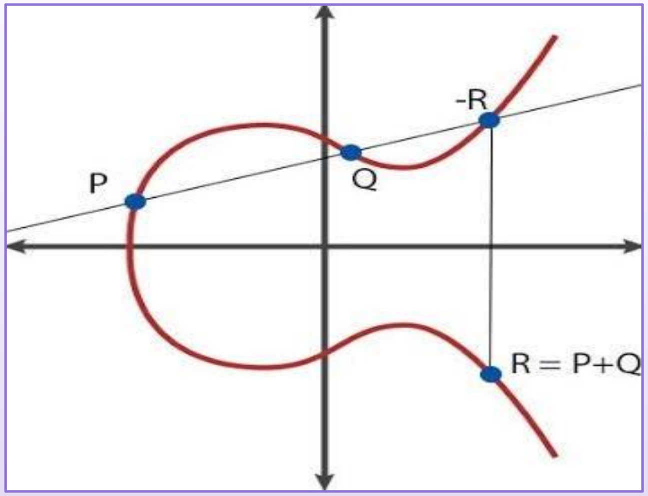
(b) $y^2 = x^3 + x + 1$

Elliptic curve: operations

- $\mathrm{P} = (x_\mathrm{P}, y_\mathrm{P}), Q = (x_Q, \mathrm{y_Q})$
- inverse
  - $-P = P' = (x_P, -y_P)$

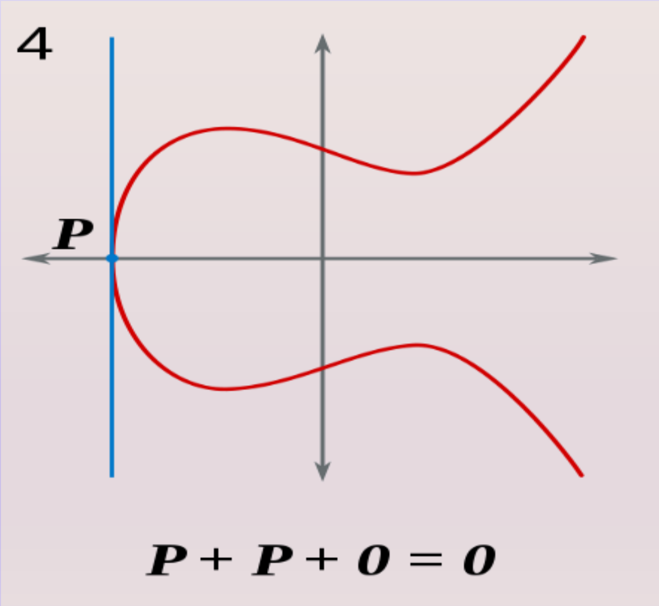


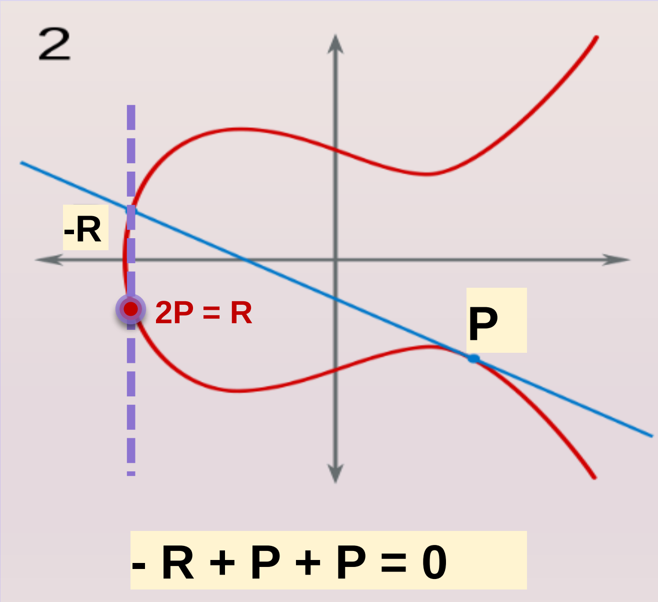
$P + P' + O = O$



- Addition of two different points
  - Case: $Q = -P$ $(x_P = x_Q, y_P = -y_Q)$
    - $P + Q = 0$
  - Case: $Q \neq -P$ $(x_P \neq x_Q)$
    - $\Delta = (y_\mathrm{Q} - y_\mathrm{P})/(x_\mathrm{Q} - x_\mathrm{P})$
    - $P + Q = R = (x_R, y_R) = \left(\Delta^2 - x_P - x_Q, \Delta(x_P - x_R) - y_P\right)$
- Consider line L: $y = y_\mathrm{P} + \Delta(x - x_\mathrm{P})$
  - Intersect E: $\left(y_\mathrm{P} + \Delta(x - x_\mathrm{P})\right)^2 = x^3 + ax + b$
  - $x^3 - \Delta^2 x^2 + \left(a - 2\Delta(y_P - \Delta x_P)\right)x + (b - (y_P - \Delta x_P)^2) = 0$
  - $(x_\mathrm{P}, y_\mathrm{P}), (x_\mathrm{Q}, y_\mathrm{Q})$ are two roots and third root $-R = (\mathrm{x}', y')$
    - $x_P + x_Q + x' = \Delta^2 \Longrightarrow x' = \Delta^2 - x_P - x_Q$
    - $\mathrm{y}' = y_P + \Delta(x' - x_P)$
    - $R = (x_R, y_R) = (\Delta^2 - x_P - x_Q, y' \ = \Delta\left(x_p - x_R\right) - y_P)$

- Double
  - Case: $y_\mathrm{P} = 0 \Longrightarrow P + P = 2P = O$
  - Case $y_\mathrm{P} \neq 0$
    - $\Delta = {}^{3x_P^2 + a}\!/_{2y_P}$
    - P+$P = 2P = R = (x_R,\ y_R) = (\Delta^2 - 2x_P, \Delta(x_P - x_R) - y_P)$





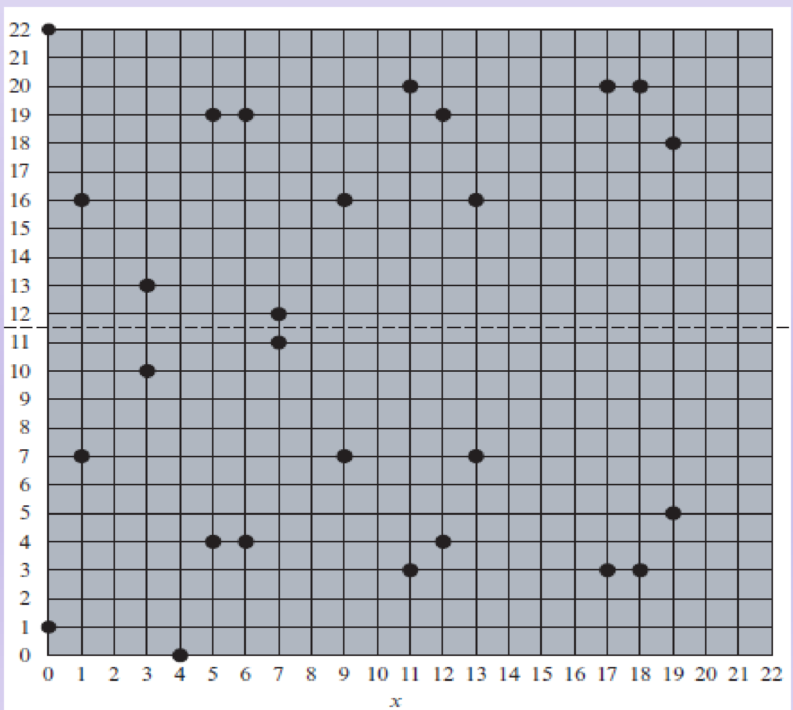
Elliptic curve over Z_p

- Two types of curves
  - prime curves: over Z_p
  - binary curves: over $GF(2^m)$
- $E_p(a,b)$: $y^2 = x^3 + ax + b$ (mod p), where $4a^3 + 27b^2 \neq 0$
- Group points : identity O and all integer points over $E_p(a,b)$
- Addition: $P = (x_{\mathrm{P}}, y_{\mathrm{P}}), Q = (x_{\mathrm{Q}}, y_{\mathrm{Q}})$
  - If $P = -Q, P + Q = O$
  - If $P \neq -Q$
    - $\lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} & if\ P \neq Q \\ \frac{3x_P^2 + a}{2y_P} & if\ P = Q \end{cases}$
    - $P + Q = (x_R, y_R) = \left(\lambda^2 - x_P - x_Q, \lambda(x_P - x_R) - y_P\right)$

Example: $E_{23}(1,1)$

- $y^2 = x^3 + x + 1 \pmod{23}$
- Group points
  - O -- identity
  - Points on the right

(0, 1)	6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)



- $P = (3, 10), Q = (9, 7)$
  - $P + Q = (x_R, y_R)$
    - $\lambda = \frac{7-10}{9-3} \bmod 23 = 11$
    - $x_R = 11^2 - 3 - 9 \bmod 23 = 17$
    - $P + Q = (17, 11(3 - 17) - 10) = (17,\ 20)$
  - $2P = (x_R, y_R)$
    - $\lambda = \frac{3\times 3^2+1}{2\times 10} \bmod 23 = 6$
    - $x_R = 6^2 - 3 - 9 \bmod 23 = 7$
    - $2P = (7,\ 6(3 - 7) - 10) = (7, 12)$

ECC: hard problem

- EC discrete logarithm problem
  - Given Q and P, compute k for the equation $Q = kP$
- No efficient algorithms for solving the EC discrete logarithm problem are known yet

EC-DH key exchange

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

EC-ElGamal encryption

- Global parameters: (E, G)
  - E is a suitable curve E, e.g. NIST P-256, P-384
  - G is a base point with large order n
- User Alice
  - private key: $PR_A = n_A, n_A < n$
  - public key: $PU_A = n_A G$
- Encryption: m
  - encode m as a point P_m in E
  - choose a random positive integer $k < n$
  - compute $C = (kG, P_m + kPU_A) = (C_1, C_2)$
- Decryption
  - compute $P_m = C_2 - PR_A C_1 = C_2 - n_A C_1$
  - decode P_m to m

EC-ElGamal encryption

- $q = 257, E_q(a, b) = E_{257}(0, -4)$
- $G = (2, 2)$
- Alice
    - $PR_A = n_A = 101$
    - $PU_A = n_A G = (197,167)$
- $P_m = (116, 26)$
- Encryption
    - Choose $k = 41$, $kG = (136, 128), kPU_A = (68, 84)$
    - $(C_1, C_2) = (kG, P_m + kPU_A) = ((136, 128), (246, 174))$
- Decryption
    - $P_m = C_2 - n_A C_1 = (246, 174) - 101(136,128) = (116, 26)$

P-256

- p: the underlined group Z_p
- h: always 1
- used in EC-DSA

Name	Value
p	0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff
a	0xffffffff00000001000000000000000000000000fffffffffffffffffffffffc
b	0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
G	(0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296, 0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5)
n	0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551
h	0x1

Key size comparison

- L: length of public key
- N: length of private key

Symmetric Key Algorithms	Diffie–Hellman, Digital Signature Algorithm	RSA (size of n in bits)	ECC (modulus size in bits)
80	$L = 1024$ $N = 160$	1024	160–223
112	$L = 2048$ $N = 224$	2048	224–255
128	$L = 3072$ $N = 256$	3072	256–383
192	$L = 7680$ $N = 384$	7680	384–511
256	$L = 15{,}360$ $N = 512$	15,360	512 +