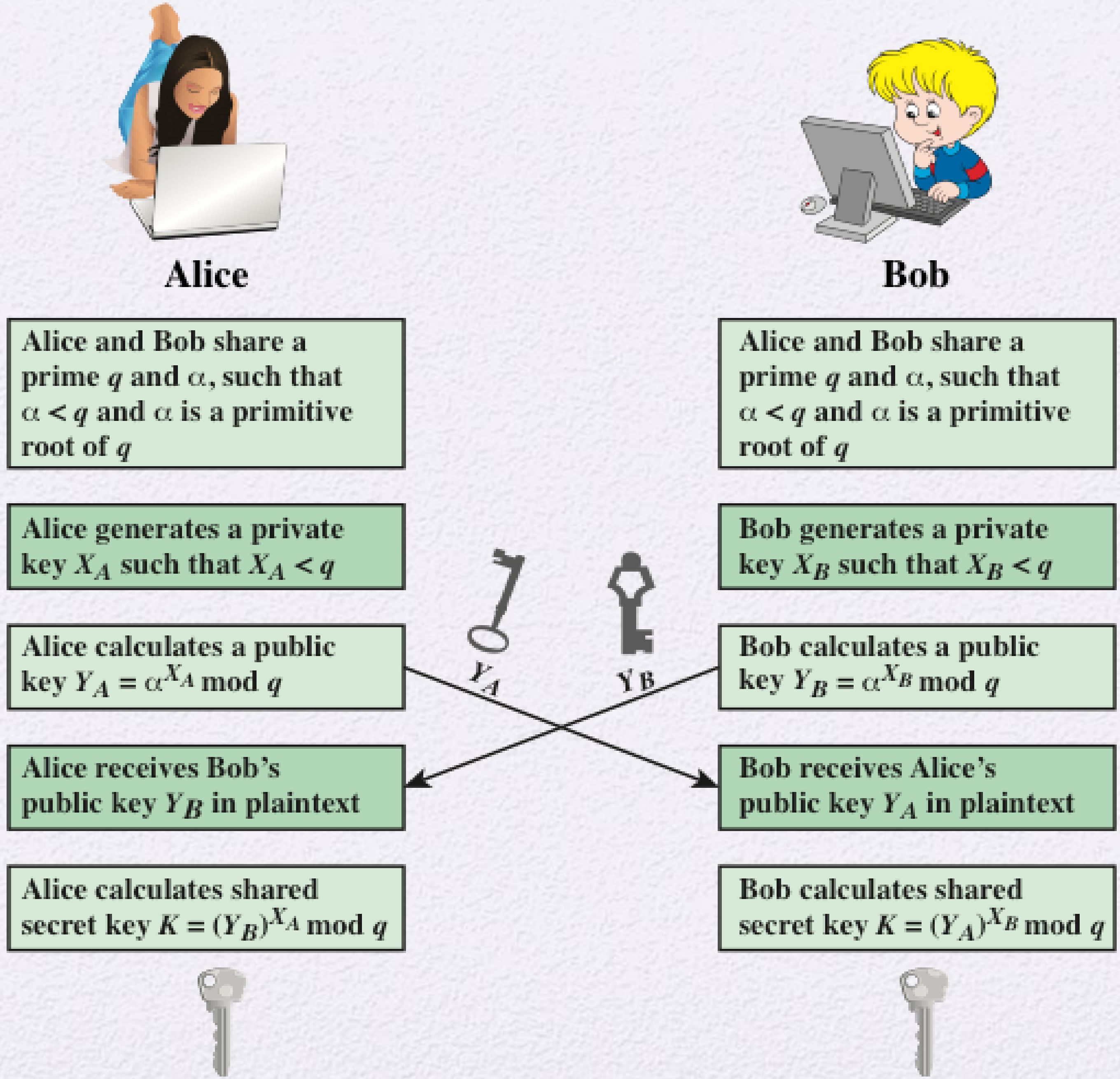
Chapter 10

Other Public-Key Cryptosystems

Diffie-Hellman Key Exchange

- First published public-key algorithm, 1976
- A number of commercial products employ this key exchange technique
- Purpose: enable two users to securely exchange a secret key over a public channel.
- Security: depend on the difficulty of computing discrete logarithms


Alice
Bob
Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q
Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q
Alice generates a private key X_A such that $X_A < q$
Bob generates a private key X_B such that $X_B < q$
Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$
Y_A
Y_B
Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$
Alice receives Bob's public key Y_B in plaintext
Bob receives Alice's public key Y_A in plaintext
Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$
Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$

DH-Key Exchange: Example

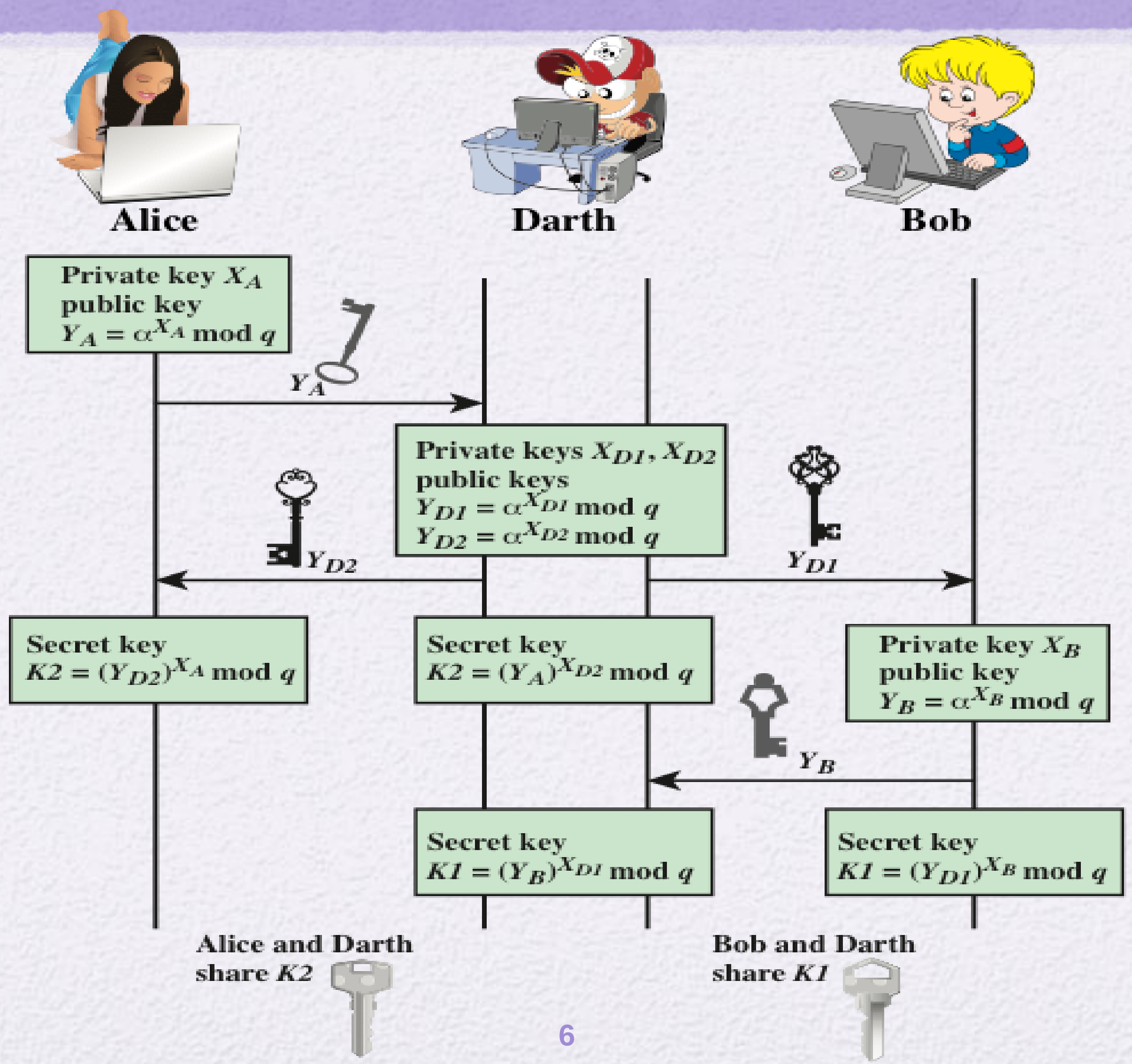
- Global parameter: $q=353$, $\alpha=3$

1. Alice: choose $X_A=97$, compute $Y_A=3^{97}$ mod 353=40, send Y_A to Bob
2. Bob: choose $X_B=233$, compute $Y_B=3^{233}$ mod 353=248, send Y_B to Alice
3. Alice: compute $K = Y_B^{X_A} = 248^{97} \bmod 353 = 160$
4. Bob: compute $K = Y_A^{X_B} = 40^{233} \bmod 353 = 160$

DH-Key Exchange: Security

- Given Y_A and Y_B, compute K
  - the DH problem
  - Still unsolvable
- DH problem is no harder than Dlog problem
  - Solving DL problem → solving DH problem
  - However the vice versa is not known yet.
- Communication: the man-in-the-middle attack

Man-in-the-Middle Attack



ElGamal Cryptography

- Taher ElGamal, 1984
- Public-key encryption and digital signature
- Security is based on the difficulty of computing discrete logarithm

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

ElGamal Encryption

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: $q=19$, $\alpha=10$
- Alice's key generation:
  - Choose $X_A=5$, compute $Y_A=10^5 \bmod 19=3$
  - $PU=(q, \alpha, Y_A)=(19, 10, 3)$, $PR=(q, \alpha, X_A)=(19, 10, 5)$
- Encryption: $M=17$, $PU=(19, 10, 3)$
  - Choose $k=6$, compute $C=(10^6 \bmod 19, 17 \times 3^6 \bmod 19)=(11, 5)$
- Decryption: $C=(11, 5)$, $PR=(19, 10, 5)$
  - Compute $M = 5/(11^5 \bmod 19) \bmod 19 = 5/7 \bmod 19$
    $= 5 \times 11 \bmod 19 = 17$

ElGamal Encryption: Security

- $X_A = \mathrm{dlog}_{q,\alpha} Y_A$: discrete logarithm problem
- $C_1, Y_A \rightarrow \alpha^{kX_A} \bmod q$: DH problem

ElGamal Encryption: Problem

- Slow
  - Encryption: two modular exponentiation
  - Decryption: one modular exponentiation
- Ciphertext expansion
  - $|C|=|C_1|+|C_2| = 2|M|$

Elliptic Curve Cryptography

- RSA
  - The key length for secure RSA has increased over recent years
  - The modulus n = 2048-bit
  - More computing time
- Elliptic curve cryptography (ECC), IEEE P1363 Standard for Public-Key Cryptography
  - Shorter key with equal security to RSA
  - Shorter key, faster: suitable for IoT devices

Abelian Group

- (G, •) is an Abelian group if

(A1) Closure: If a and b belong to G, then $a \bullet b$ is also in G

(A2) Associative: $a \bullet (b \bullet c) = (a \bullet b) \bullet c$ for all a, b, c in G

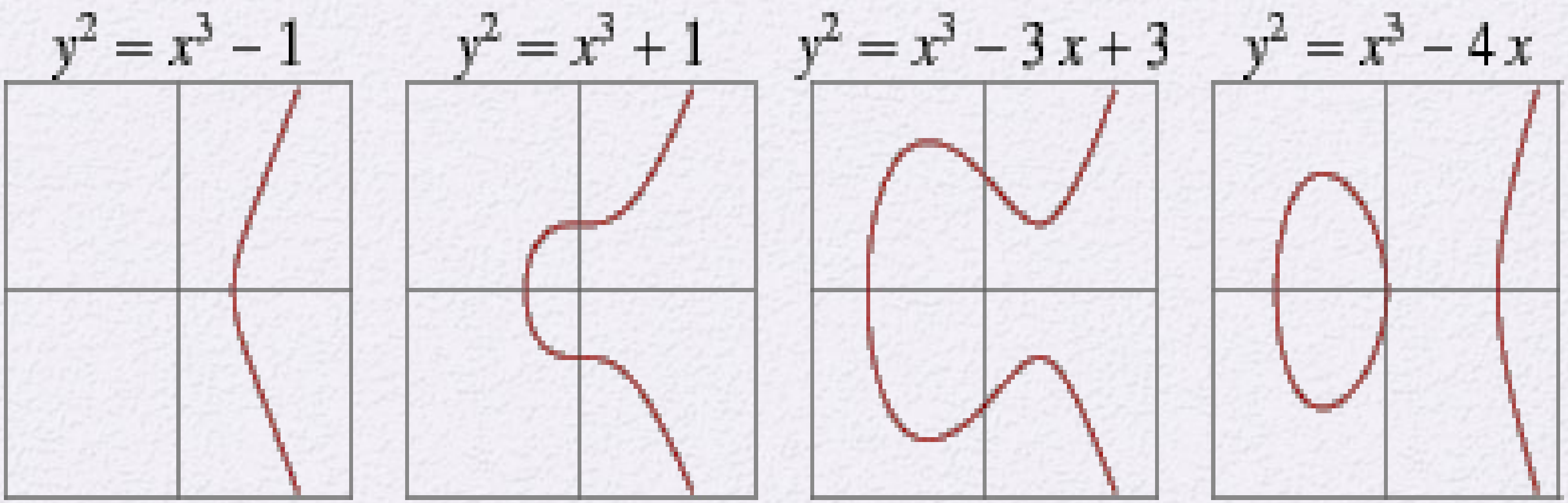
(A3) Identity element: There is an element e in G such that $a \bullet e = e \bullet a = a$, for all a in G

(A4) Inverse element: For each a in G, there is an element a' in G such that $a \bullet a' = a' \bullet a = e$

(A5) Commutative: $a \bullet b = b \bullet a$ for all a, b in G

Elliptic Curves over Reals

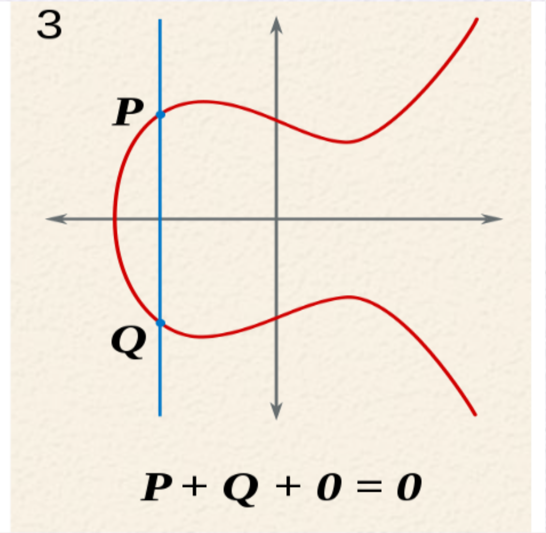
- Weierstrass equation: E: $y^2 = x^3 + ax + b$
  - $4a^3+27b^2 \neq 0$: non-singular
- Example



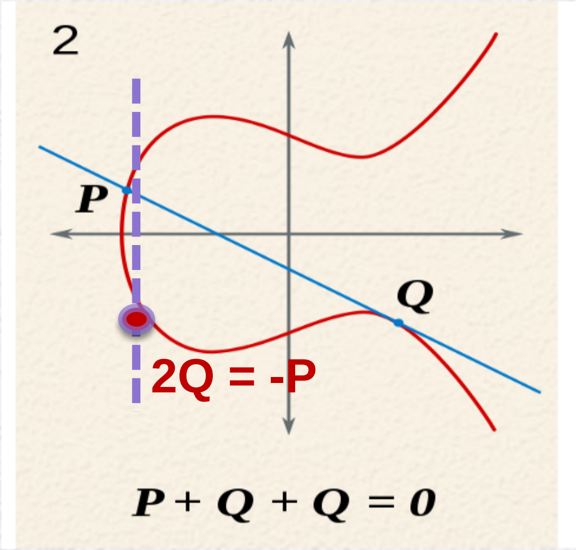
EC → Additive Group

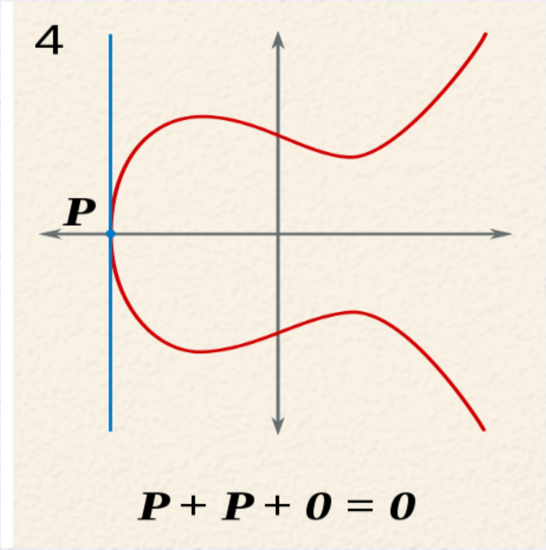
- G = points over E U {O}
- O: zero point -- infinity
- **Define "+" on G**
  - **P+Q+R=O for P, Q, R in a line**
- Operations:
  - inverse
  - double
  - addition

- Inverse: P=(x, y) ➔ -P = (x, -y) -- in (3)

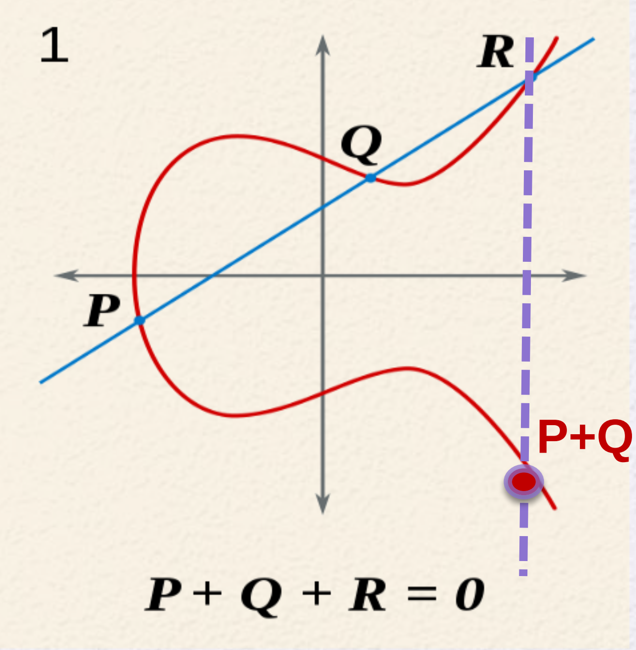


- Double: 2(x, y)
  - Case I: y = 0. Like P=(x, 0) in (4),
    - 2P = 0
  - Case II : y≠0. Like Q=(x, y) in (2),
    - 2Q = (x’, y’) = $\left(\left(\frac{3x^2+a}{2y}\right)^2 - 2x,\ \frac{3x^2+a}{2y}(x - x') - y\right)$
  - kP = P+P+…+P -- k-1 additions





- Addition: $P=(x_1, y_1)$, $Q=(x_2, y_2)$
  - If $Q=-P$, $Q+P=O$
  - If $Q \neq -P$, $P + Q = (x_3, y_3) = (\Delta^2 - x_1 - x_2,\ -y_1 + \Delta(x_1 - x_3))$ where $\Delta=(y_2-y_1)/(x_2-x_1)$ -- in (1)



Consider the line $y=y_1+\Delta(x-x_1)$

Intersection L with E: $y^2=x^3+ax+b$

$$(y_1+\Delta(x-x_1))^2 = x^3+ax+b$$

$\Rightarrow$ $x^3 - \Delta^2 x^2 + (a-2\Delta(y_1-\Delta x_1))\, x + (b-(y_1-\Delta x_1)^2 = 0$

$\Rightarrow$ x_1, x_2 are two roots, the third root x_3

$\Rightarrow$ $x_1+x_2+x_3 = \Delta^2$

$\Rightarrow$ $x_3 = \Delta^2 - x_1 - x_2$

Elliptic Curves over Z_p

- $E_p(a, b)$: $y^2 = x^3 + ax + b$ (mod p), where $4a^3+27b^2 \neq 0$
- Group points
  - O -- identity
  - All integer points over $E_p(a, b)$
- Operation: addition, $P=(x_P\ y_P)$, $Q=(x_Q, y_Q)$
  - $P+O = P$, $-P = (x_p, -y_P)$, $aP = P+P+\ldots+P$
  - If $P \neq -Q$, $P+Q = (x_R, y_R) = (\lambda^2-x_P-x_Q,\ \lambda(x_P-x_R)-y_P)$, where

$$\lambda = \begin{cases} \frac{y_Q-y_P}{x_Q-x_P} & if\ P \neq Q \\ \frac{3x_P^2+a}{2y_P} & if\ P = Q \end{cases}$$

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

- Group points
  - O -- identity
  - Points over $E_{23}(1, 1)$

(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)

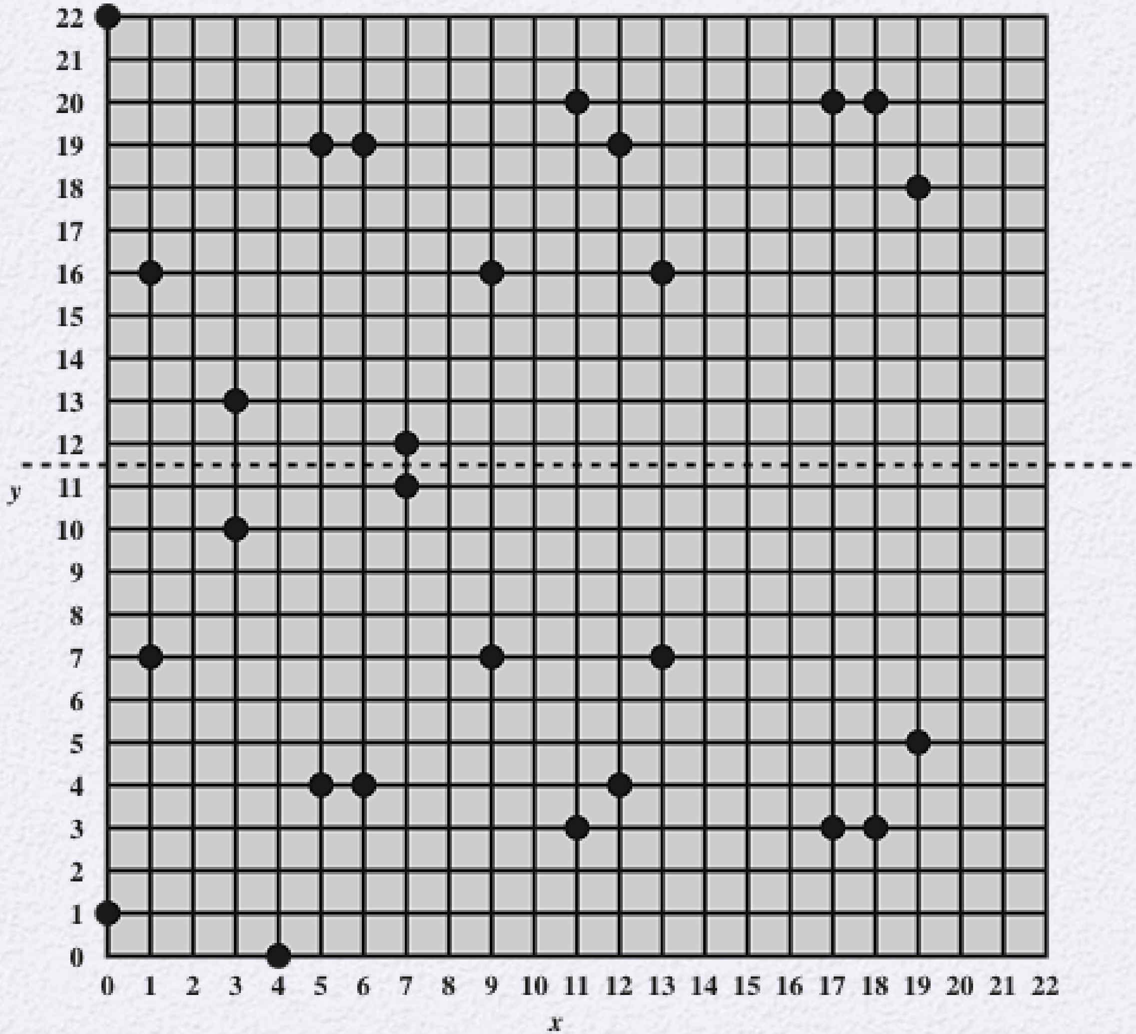


Figure 10.5 The Elliptic Curve $E_{23}(1,1)$

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

- $P = (3, 10)$, $Q = (9, 7)$
  - $P+Q = (x_3, y_3)$
    - $\lambda = (7-10)/(9-3) = 11,$
    - $P+Q = (11^2-3-9, 11(3-17)-10) = (17, 20)$
  - $2P = (x_3, y_3)$
    - $\lambda = (3x3^2+1)/(2x10) = 6,$
    - $2P = (6^2-3-3, 6(3-7)-10) = (7, 12)$
  - $4p=2(2P) = 2(7,12) = ?$

ECC Hard Problem

- Given k and P, it is easy to compute kP by the double-and-add problem
- EC logarithm problem
  - Given Q and P, find k for the equation Q=kP

ECDH: 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

- Select suitable curve E and point G as in Diffie-Hellman
- First encode the message *m* as a point P_m on the EC
- Each user chooses a private key n_A and generates a public key $P_A = n_A G$
- Encrypt: for message P_m,
  - choose a random positive integer *k*
  - compute ciphertext $C_m = \{P_B = kG, P_m + kP_A\}$
- Decrypt: for ciphertext $C_m = \{P_B, P_m + kP_A\}$, compute

$$P_m + kP_A - n_A P_B = P_m + k(n_A G) - n_A(kG) = P_m$$

EC-ElGamal: Security

- Depend on the difficulty of the elliptic curve logarithm problem
  - Given P and Q =kP, it is difficult compute k.
- Fastest known technique for EC logarithm problem is "Pollard rho method"
- Compared to factoring, can use much smaller key sizes than RSA
- For equivalent key lengths, computations are roughly equivalent
- Hence, for similar security, ECC offers significant computational advantages over RSA

Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis (NIST SP-800-57)

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+

Note: L = size of public key, N = size of private key

PRNG Based on Asymmetric Cipher

- Use existent asymmetric encryption algorithm
- Security is provable based on the difficulty problem of the chosen algorithm
- Much slower. It is used to generate open-ended PRNG bit streams
- Useful for creating a pseudorandom function (PRF) for generating a short pseudorandom bit sequence

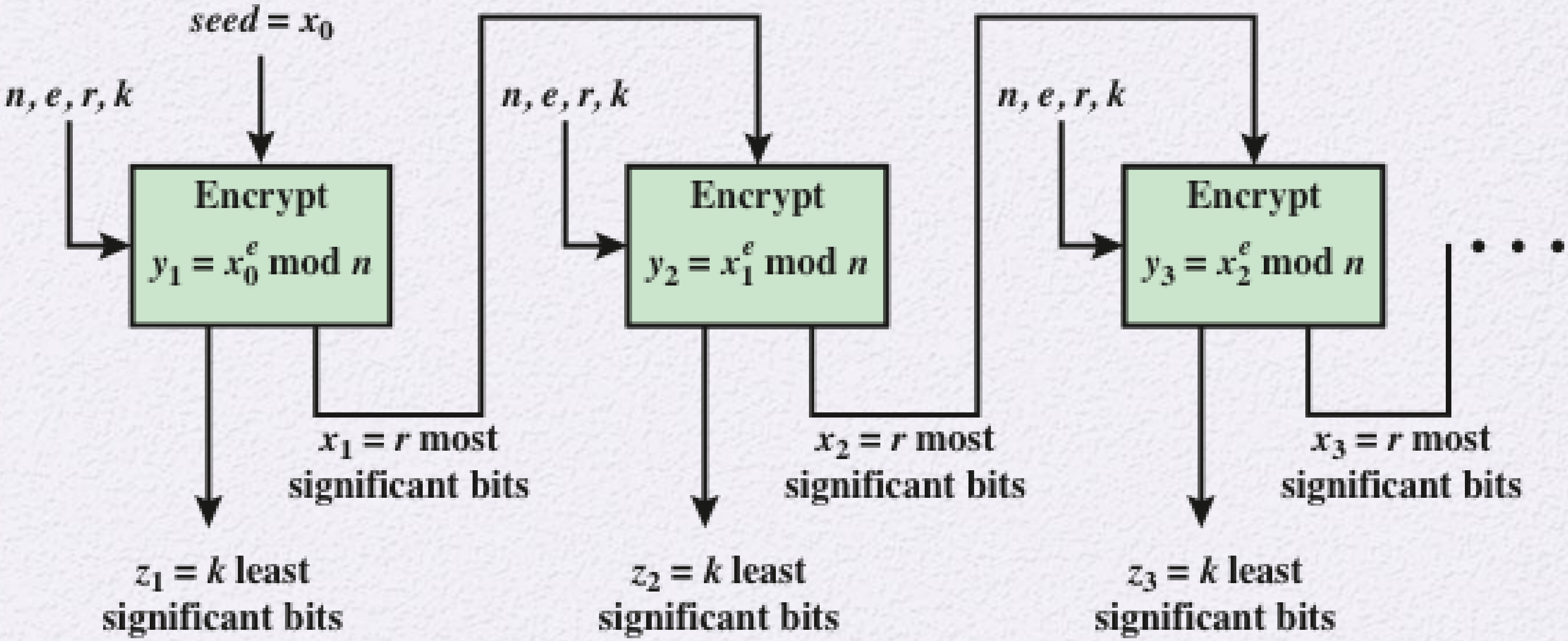


Figure 10.8 Micali-Schnorr Pseudorandom Bit Generator

PRNG Based on ECC

- U.S. National Security Agency (NSA)
- Dual elliptic curve PRNG (DEC PRNG)
  - P, Q: two points, s_0: random seed
  - $x(P)$ = the x-coordinate of point P
  - For i=1 to k do

    set $s_i = x(s_{i-1}P)$

    set $r_i = lsb_{240}(x(s_iQ))$

    End for

    Return $(r_1\ r_2 \ldots\ r_k)$
- Recommended in NIST SP 800-90, the ANSI standard X9.82, and the ISO standard 18031

Summary

- Diffie-Hellman Key Exchange
  - The algorithm
  - Key exchange protocols
  - Man-in-the-middle attack
- Elgamal cryptographic system
- Elliptic curve cryptography
  - Analog of Diffie-Hellman key exchange
  - Elliptic curve encryption/decryption
  - Security of elliptic curve cryptography
- Elliptic curve arithmetic
  - Abelian groups
  - Elliptic curves over real numbers
  - Elliptic curves over Z_p
  - Elliptic curves over $GF(2^m)$
- Pseudorandom number generation based on an asymmetric cipher
  - PRNG based on RSA
  - PRNG based on elliptic curve cryptography