Chapter 10

Other Public-Key Cryptosystems

Diffie-Hellman Key Exchange

- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- Purpose is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages
- The algorithm itself is limited to the exchange of secret values
- Its effectiveness depends on the difficulty of computing discrete logarithms

Publicly known numbers:
- **prime number q**
- **integer α that is a primitive root of q .**

Example:
$q=97$
$\alpha=5$

Alice

Bob

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 $Y_A = \alpha^{X_A} \bmod q$	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 $K = (Y_A)^{X_B} \bmod q$

Y_A Y_B

Private Keys: X_A and X_B. → $X_A=36$, $X_B=58$

Public Keys: Y_A and Y_B. → $Y_A=50$, $Y_B=44$

$K=75$ $K=75$

Share Key have Same values with Bob and Alice

$$K= (Y_B)^{X_A} \bmod q \\ =(\alpha^{X_B})^{X_A} \bmod q \\ =\alpha^{X_B X_A} \bmod q$$

$$K= (Y_A)^{X_B} \bmod q \\ =(\alpha^{X_A})^{X_B} \bmod q \\ =\alpha^{X_A X_B} \bmod q$$

Figure 10.1 Diffie-Hellman Key Exchange

(Reading)

DH key exchange protocol is vulnerable to Man-in-Middle attack because it does not authenticate the participants.
This vulnerability can be overcome with the use of digital signatures and public- key certificates.

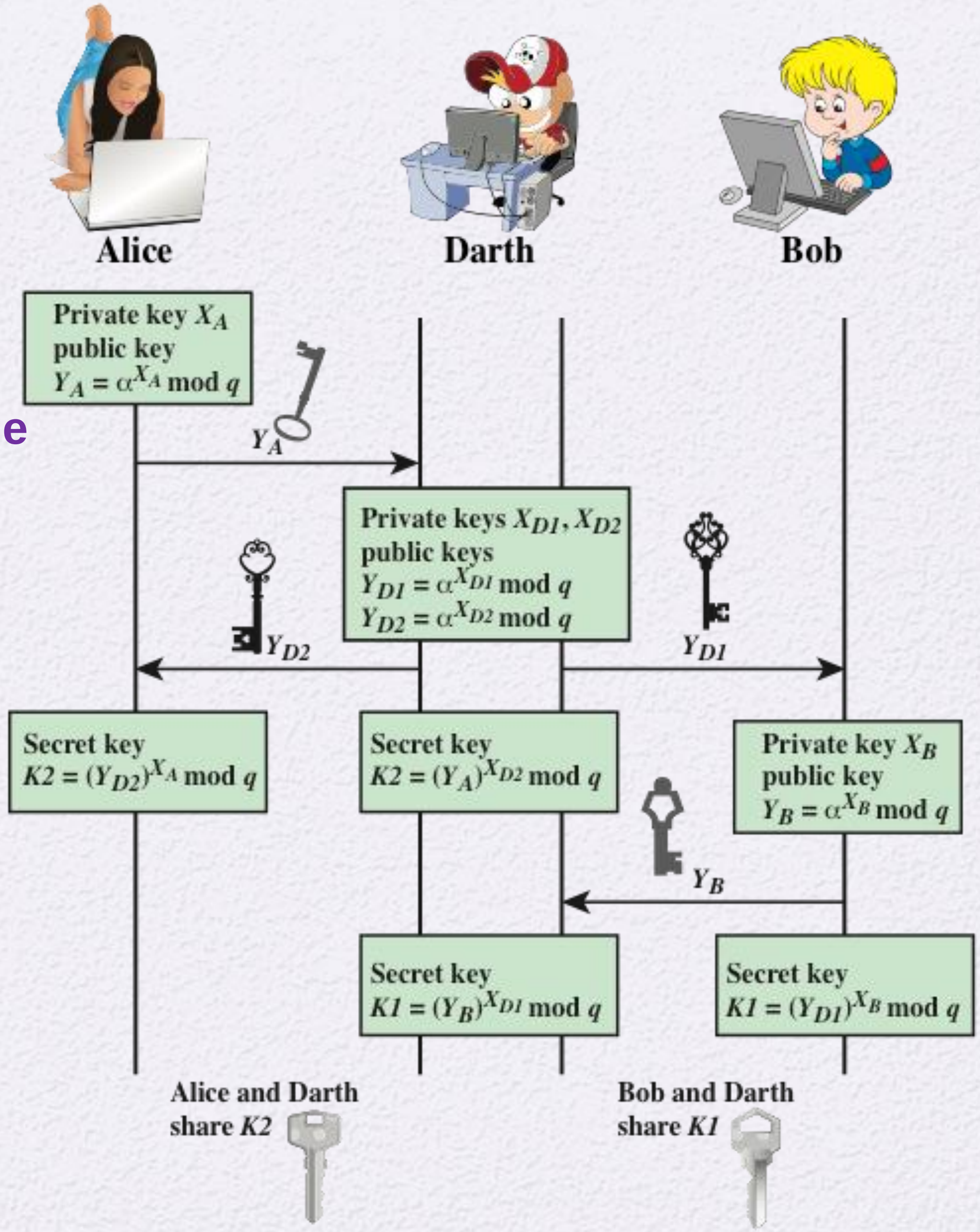


Figure 10.2 Man-in-the-Middle Attack

ElGamal Cryptography طاهر الجمل

- Taher ElGamal is an Egyptian cryptographer who proposed:
  - ElGamal encryption: an asymmetric key encryption algorithm for public-key encryption.
  - ElGamal signature scheme, a digital signature scheme.
- ElGamal encryption is a public-key scheme:
  - based on discrete logarithms,
  - closely related to the Diffie-Hellman technique
  - Global elements of ElGamal are a prime number q and a, which is a primitive root of q.
  - User A generates a private/public key pair.
- The security of ElGamal is based on the difficulty of computing discrete logarithms, to recover keys.

ElGamal

Alice

q is prime number
α is a primitive root
$X_A < q-1$
$Y_A = \alpha^{X_A} \bmod q$

$PU=\{q, \alpha, Y_A\}$

Bob

$k < q$
$M < q$

$K = \alpha^{X_A\ k} \bmod q$
$C_1 = \alpha^k \bmod q$
$C_2 = M\ K \bmod q$

$Cipher = \{C_1, C_2\}$

$K = (C_1)^{XA} \bmod q$
$\quad = (\alpha^k)^{XA} \bmod q$
$M = (C_2\ K^{-1}) \bmod q$

Y_A (Alice➔Bob) $\qquad$ C_1 (Bob➔Alice)

$$K = \alpha^{XA\ k} \bmod q = \alpha^{XA\ k} \bmod q = \alpha^{k\ XA} \bmod q$$

from Alice $\qquad$ from Bob

Proof for AlGamal Encryption

Need to show the K and M computed by Alice are same K and M used by Bob.

First: start with K at Alice side
$K = \alpha^{X_A \times k} \mod q$

$K = (C_1)^{XA} \mod q$
$= (\alpha^k)^{XA} \mod q$
$= (\alpha^{XA})^k \mod q$
$= Y_A^k \mod q$
$= K$ at Bob side

Second: M at Alice side
Need to prove that M recovered by Alice is Same M encrypted by Bob.
$M = (C_2 K^{-1}) \mod q$
$= M K K^{-1} \mod q$
$= M$

Example

Assume:
$q=107$
$\alpha=2$

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	$Y_A = \alpha^{X_A} \mod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

$X_A=67$
$Y_A=94$

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

(*k* is private Key for Bob)

k=45
M=66
K=5
C1=28
C2=9
C=(28,9)

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

K=5
$K^{-1}=43$
M=66

AlGamal Encryption

Elliptic Curve Cryptography (ECC)

- Why ECC?
- Elliptic Curves over real numbers
- Elliptic curves over Z_p : *i.e. GF(p)*
- Elliptic Curves over $GF(2^m)$
- ECC: Key Exchange and Encryption

Why Elliptic Curve Cryptography (ECC)?

- Why ECC?
  - The key length for secure RSA use has increased over recent years resulting in heavier processing load.
  - ECC provides equivalent level of security with **smaller keys**.
    - ECC with Key size of **256-bit** ≈ RSA with key size of **3072-bits**
    - ECC with Key size of **324-bit** ≈ RSA with key size of **7680-bits**
- Elliptic curve cryptography (ECC) is the IEEE P1363 Standard for Public-Key Cryptography

Parameters	**ECC**	**RSA**
Computational Overheads	Roughly 10 times than that of RSA can be saved	More than ECC
Key Sizes	System parameters and key pairs are shorter for the ECC.	System parameters and key pairs are larger for the RSA.
Bandwidth saving	ECC offers considerable bandwidth savings over RSA	Much less bandwidth saving than ECC
Key Generation	Faster	Slower
Encryption	Much Faster than RSA	At good speed but slower than ECC
Decryption	Slower than RSA	Faster than ECC
Small Devices efficiency	Much more efficient	Less efficient than ECC

Comparison ECC vs. RSA

Source:
V.Kute et al,"A software comparison of RSA & ECC",
R. Sinha et al "Performance Based Comparison Study of RSA and Elliptic Curve Crypto

Types of ECC we will discus

Elliptic Curve Cryptography

Elliptic Curves over real numbers

Elliptic curves over Field

Elliptic curves over Z_p

Elliptic Curves over $GF(2^m)$

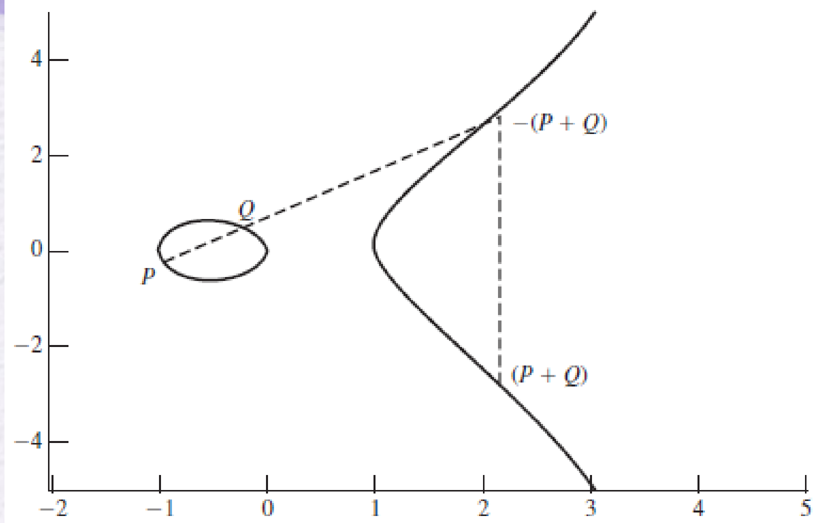
Parameter	Definition
Zp: $\{0, 1, \ldots, p\}$ or $GF(2^m)$	Base Field
a , b	Coefficients of elliptic curve
G	Generator: a base point that satisifies elliptic equation
n	Order of G: $n \times G = O$; n is a prime

1) Elliptic Curves over Real Numbers

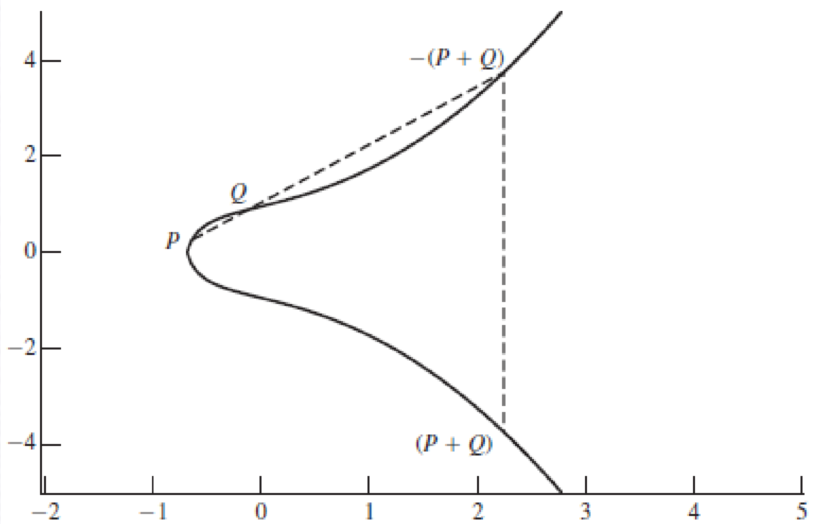
- cubic equations for elliptic curves take the following form, known as a **Weierstrass equation:**
  - $y^2 + axy + by = x^3 + cx^2 + dx + e$
  - where a, b, c, d, e are real numbers and x and y take on values in the real numbers
- We will limit ourselves to equations of the form:
  - $y^2 = x^3 + \boldsymbol{a}x + \boldsymbol{b}$

Elliptic Curves over Real Numbers

- **E(*a*, *b*)** consisting of all of the points (x, y) that satisfy Equation
  - $y^2 = x^3 + ax + b$
- Using this terminology, the two curves in Figures depict the sets E(-1, 0) and E(0, 1), respectively.



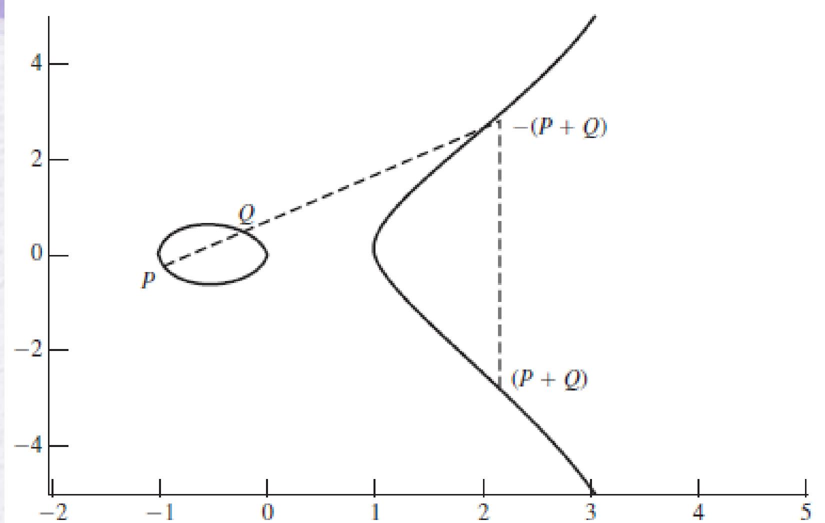
(a) $y^2 = x^3 - x$



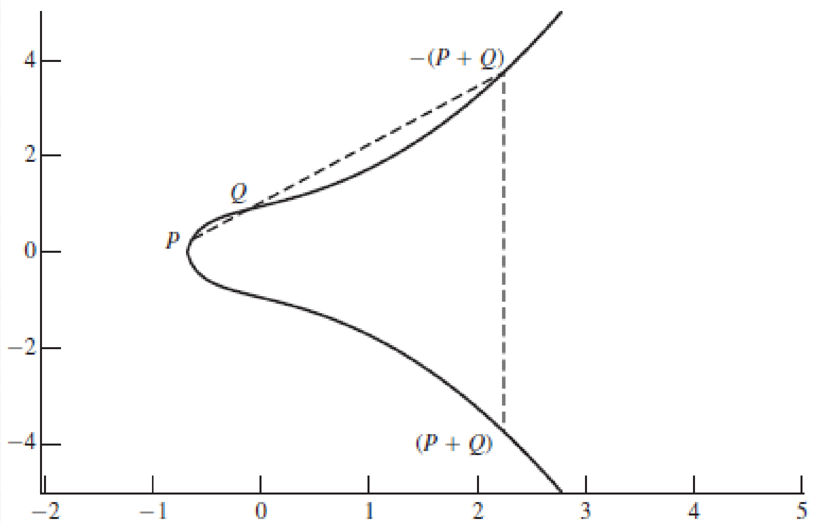
(b) $y^2 = x^3 + x + 1$

Elliptic Curves over Real Numbers: Addition

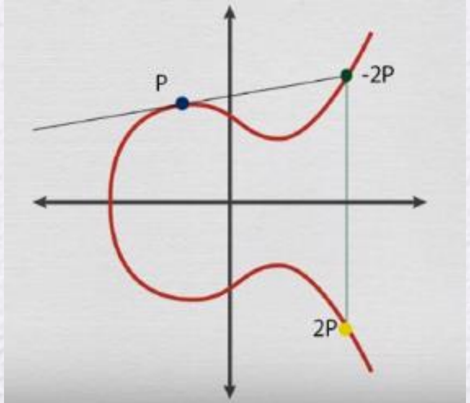
- $\mathcal{O}$ called **the point at infinity** or **the zero point**
- $\mathcal{O}$ serves as the **additive identity**, properties:
  1. $\mathcal{O} = -\mathcal{O}$
  2. $P + \mathcal{O} = P$
  3. $P + (-P) = P - P = \mathcal{O}$
- The negative of a point $P = (x, y)$ is $-P = (x, -y)$
- Addition: adding points **P** and **Q** with different x coordinates, draw a straight line between them and find the third point of intersection **R.**
  - P + Q = -R.
  - P + Q to be the mirror image (with respect to the x axis) of the third point of intersection.
- To double a point Q, draw the tangent line and find the other point of intersection S. Then Q + Q = 2Q = -S



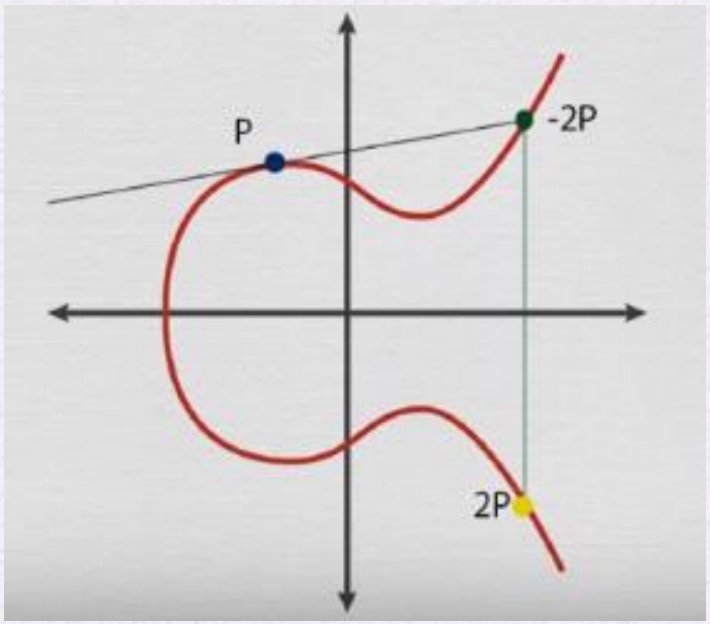
(a) $y^2 = x^3 - x$

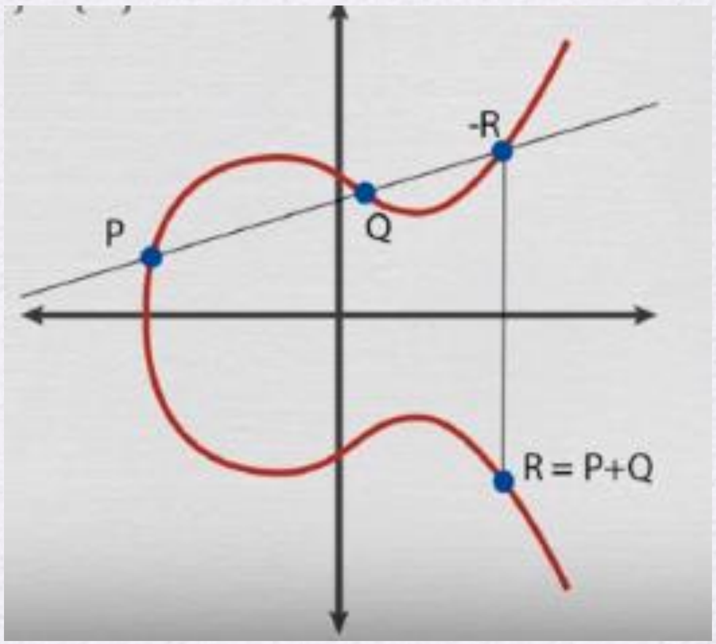


(b) $y^2 = x^3 + x + 1$



Group Operations: Addition “+”, point doubling





Adding Vertical Points & Scalar Multiplication

Scalar Multiplication

$P \in E$

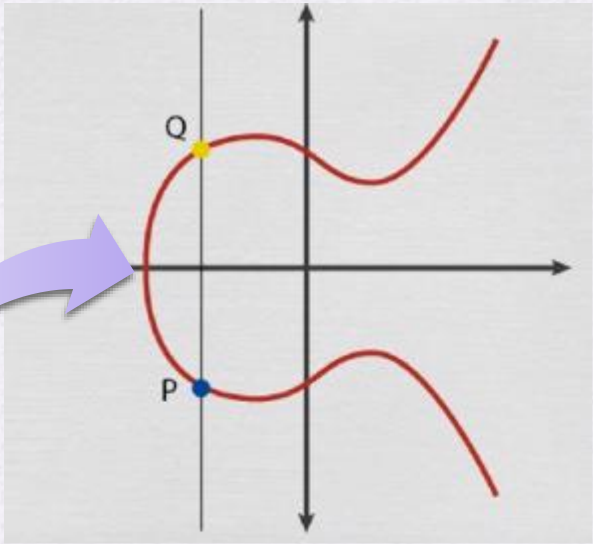
$k \in \mathbb{Z}$

$Q = kP$

REPEATED ADDITION

$Q = P + P + \ldots + P$ } K times

Adding Vertical Points



$P + Q = \mathcal{O}$ if $x_P = x_Q$

$P + P = \mathcal{O}$ if $y_P = 0$

Elliptic Curves over Real Numbers: Addition

- For two distinct points, $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$
- Slope of the line *connecting two points* :
  - $\Delta = \lambda = s = (y_Q - y_P)/(x_Q - x_P)$
- There is exactly one other point intersects the elliptic curve, and that is the negative of the sum of *P* and *Q*.
- Computing coordinates of : **$R = P + Q$**

$$\begin{aligned} x_R &= \Delta^2 - x_P - x_Q \\ y_R &= -y_P + \Delta(x_P - x_R) \end{aligned} \tag{10.3}$$

We also need to be able to add a point to itself: $P + P = 2P = R$. When $y_P \neq 0$, the expressions are

$$\begin{aligned} x_R &= \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \\ y_R &= \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P \end{aligned} \tag{10.4}$$

ECC in Finite Fields: ECC Prime cures and Binary Curves

- ECC makes use of elliptic curves in which the **variables** and **coefficients** are all restricted to elements of a **finite field**.
- Two families of elliptic curves in cryptographic applications:
  1. Prime curves over finite field Z_p
     - Variables and coefficients are calculated using (**mod *p***)
     - Best for software applications
  2. Binary curves over $GF(2^m)$
     - Variables and coefficients are calculated over $GF(2^m)$
     - Best for hardware applications

Finite Field Z_p: *Quick Review*

- Z_p is set of non-negative integers: $\{0, 1, \ldots, p\text{-}1\}$
  - This is referred to as the **set of residues**, or **residue classes** (mod p). All mathematical results are applied to (mod ***p***)

The residue classes (mod 4) are

$$[0] = \{\ldots, -16, -12, -8, -4, 0, 4, 8, 12, 16, \ldots\}$$
$$[1] = \{\ldots, -15, -11, -7, -3, 1, 5, 9, 13, 17, \ldots\}$$
$$[2] = \{\ldots, -14, -10, -6, -2, 2, 6, 10, 14, 18, \ldots\}$$
$$[3] = \{\ldots, -13, -9, -5, -1, 3, 7, 11, 15, 19, \ldots\}$$

- Properties of modulo arithmetic in Zp

Property	Expression
Commutative Laws	$(w + x) \bmod n = (x + w) \bmod n$ $(w \times x) \bmod n = (x \times w) \bmod n$
Associative Laws	$[(w + x) + y] \bmod n = [w + (x + y)] \bmod n$ $[(w \times x) \times y] \bmod n = [w \times (x \times y)] \bmod n$
Distributive Law	$[w \times (x + y)] \bmod n = [(w \times x) + (w \times y)] \bmod n$
Identities	$(0 + w) \bmod n = w \bmod n$ $(1 \times w) \bmod n = w \bmod n$
Additive Inverse $(-w)$	For each $w \in Z_n$, there exists a z such that $w + z \equiv 0 \bmod n$

2) Elliptic curves over Z_p

The following are the steps to Compute Elliptic Curve points over Z_p :

- Define the valid values of *x*'s and *y*'s
  - For Z_p valid values are set of non-negative integers: $\{0, 1, \ldots, p\text{-}1\}$
- Compute G and its group: G, 2G, 3G, … , nG
  - Select Elliptic curve equation
  - Search for generator point G
  - Generate cyclic group. Every point in the sub-group can be reached by repeated addition of G point. So:
    - 2G= G+G
    - 3G= G + 2G
    - …
    - nG= $\mathcal{O}$
  - Size of the group: ord(G)=n

Elliptic curves over Z_p

- The algebraic equations of elliptic curve arithmetic over real numbers applies to Z_p
  - Coefficients and variables limited to (*mod p*)
  - $y^2 \pmod{p} = (x^3 + ax + b) \pmod{p}$

- Example: $a = 1, b = 1, x = 9, y = 7, p = 23$

$7^2 \bmod 23 = (9^3 + 1 \times 9 + 1) \bmod 23$

$49 \bmod 23 = 739 \bmod 23$

$3 = 3$

➔ Point (9, 7) is a point in this prime elliptic $E_p(a,b) = E_{23}(1,1)$

elliptic curves over Z_p : Example

- let $p = 23$ consider the elliptic curve $y^2 = x^3 + x + 1$

Table 10.1 Points (other than O) on the Elliptic Curve $E_{23}(1,1)$

(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)

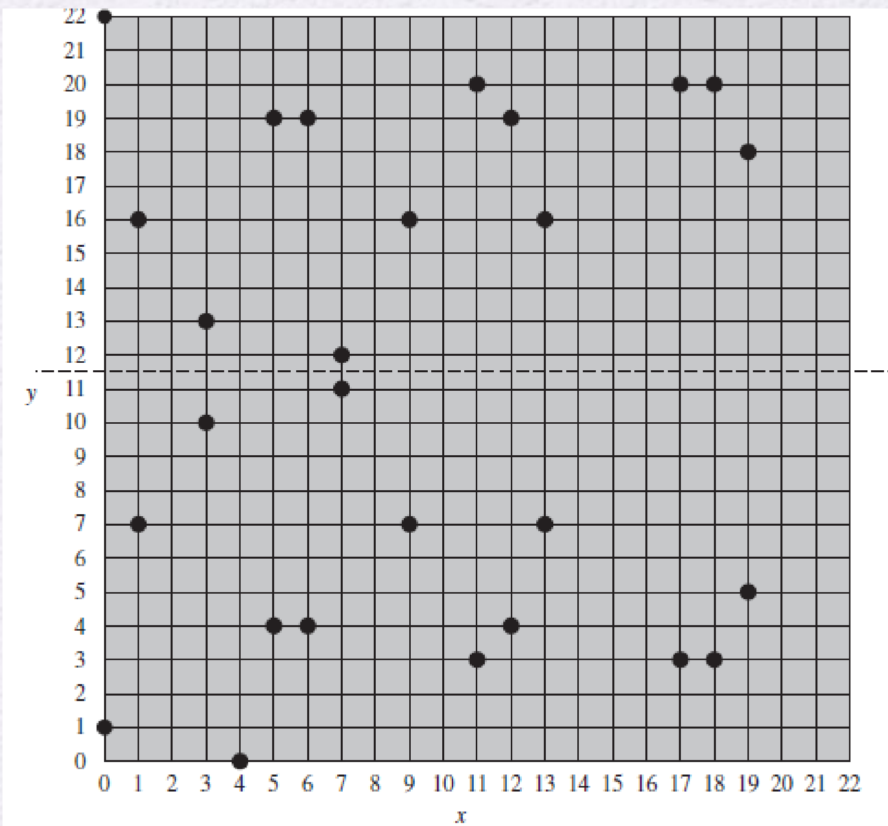


Figure 10.5 The Elliptic Curve $E_{23}(1, 1)$

elliptic curves over Z_p : Example

Table 10.1 Points (other than O) on the Elliptic Curve $E_{23}(1,1)$

(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)

Group: G -> 2G -> …… > nG	size
(0,1) → (6,19) → (3,13) → (13,16) → (18,3) → (7,11) → (11,3) → (5,19) → (19,18) → (12,4) → (1,16) → (17,20) → (9,16) → (4,0) → (9,7) → (17,3) → (1,7) → (12,19) → (19,5) → (5,4) → (11,20) → (7,12) → (18,20) → (13,7) → (3,10) → (6,4) → (0,22) → O	28
(6,19) → (13,16) → (7,11) → (5,19) → (12,4) → (17,20) → (4,0) → (17,3) → (12,19) → (5,4) → (7,12) → (13,7) → (6,4) → O	14
(5,4) → (17,20) → (13,16) → (13,7) → (17,3) → (5,19) → O	7
(11,20) → (4,0) → (11,3) → O	4
(4,0) → O	2

The rules for addition over $E_p(a, b)$

- Same as elliptic curve over real numbers.

1. $P + O = P$.
2. If $P = (x_P, y_P)$, then $P + (x_P, -y_P) = O$. The point $(x_P, -y_P)$ is the negative of P, denoted as $-P$. For example, in $E_{23}(1, 1)$, for $P = (13, 7)$, we have $-P = (13, -7)$. But $-7 \bmod 23 = 16$. Therefore, $-P = (13, 16)$, which is also in $E_{23}(1, 1)$.
3. If $P = (x_p, y_p)$ and $Q = (x_Q, y_Q)$ with $P \neq -Q$, then $R = P + Q = (x_R, y_R)$ is determined by the following rules:

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p$$
$$y_R = (\lambda(x_P - x_R) - y_P) \bmod p$$

where

$$\lambda = \begin{cases} \left(\dfrac{y_Q - y_P}{x_Q - x_P}\right) \bmod p & \text{if } P \neq Q \\[2ex] \left(\dfrac{3x_P^2 + a}{2y_P}\right) \bmod p & \text{if } P = Q \end{cases}$$

4. Multiplication is defined as repeated addition; for example, $4P = P + P + P + P$.

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p$$
$$y_R = (\lambda(x_P - x_R) - y_P) \bmod p$$

Example

$$\lambda = \begin{cases} \left(\dfrac{y_Q - y_P}{x_Q - x_P}\right) \bmod p & \text{if } P \neq Q \\[2ex] \left(\dfrac{3x_P^2 + a}{2y_P}\right) \bmod p & \text{if } P = Q \end{cases}$$

For example, let $P = (3,\ 10)$ and $Q = (9,\ 7)$ in $E_{23}(1, 1)$. Then

$$\lambda = \left(\frac{7 - 10}{9 - 3}\right) \bmod 23 = \left(\frac{-3}{6}\right) \bmod 23 = \left(\frac{-1}{2}\right) \bmod 23 = 11$$

$$x_R = (11^2 - 3 - 9) \bmod 23 = 109 \bmod 23 = 17$$

$$y_R = (11(3 - 17) - 10) \bmod 23 = -164 \bmod 23 = 20$$

So $P + Q = (17, 20)$. To find $2P$,

$$\lambda = \left(\frac{3(3^2) + 1}{2 \times 10}\right) \bmod 23 = \left(\frac{5}{20}\right) \bmod 23 = \left(\frac{1}{4}\right) \bmod 23 = 6$$

The last step in the preceding equation involves taking the multiplicative inverse of 4 in Z_{23}. This can be done using the extended Euclidean algorithm defined in Section 4.4. To confirm, note that $(6 \times 4) \bmod 23 = 24 \bmod 23 = 1$.

$$x_R = (6^2 - 3 - 3) \bmod 23 = 30 \bmod 23 = 7$$
$$y_R = (6(3 - 7) - 10) \bmod 23 = (-34) \bmod 23 = 12$$

and $2P = (7, 12)$.

Computation of Fractions in mod q

$$\lambda = \left(\frac{7-10}{9-3}\right) \bmod 23 = \left(\frac{-3}{6}\right) \bmod 23 = \left(\frac{-1}{2}\right) \bmod 23 = 11$$

How?

- Step 1: apply mod on numerator and denominator (if larger than ***p***), then simplify fraction.
- Step 2: multiply with multiplicative inverse of denominator
- Step 3: check your answer
- Example 1:

  (-3/6) mod 23 = (-1/2) mod 23 ; Multiplicative inverse of 2 is 12

  (-1/2) mod 23 ≡ (-1×12) mod 23 ≡ -12 mod 23 ≡ 11

  Check your answer: (11 ×2) mod 23 = 22 = -1

- Example 2:

  (28/20) mod 23 ≡ (5/20) mod 23 ; apply (mod 23) on numerator

  (5/20) mod 23 ≡ (1/4) mod 23 ; multiplicative invers of 4 is 6

  (1/4) ≡ (1 × 6) mod 23 ≡ 6

  Check your answer: (20 ×6) mod 23 ≡ 5 ≡ 28 mod 23

Elliptic curves over $(mod\ q)$

Example (2): $y^2=x^3+2x+2$, p=17, n=19

$$E: \; y^2 \equiv x^3 + 2x + 2 \pmod{17}$$

$G = (5,1)$	$11G = (13,10)$
$2G = (6,3)$	$12G = (0,11)$
$3G = (10,6)$	$13G = (16,4)$
$4G = (3,1)$	$14G = (9,1)$
$5G = (9,16)$	$15G = (3,16)$
$6G = (16,13)$	$16G = (10,11)$
$7G = (0,6)$	$17G = (6,14)$
$8G = (13,7)$	$18G = (5,16)$
$9G = (7,6)$	$19G = \mathcal{O}$
$10G = (7,11)$	

$n = 19$

$h = 1$

Illustration of computing 2G

COMPUTE $2G = G + G$

$$s = \frac{3x_G^2 + a}{2y_G} \qquad s \equiv \frac{3(5^2)+2}{2(1)} \equiv 77 \cdot 2^{-1} \equiv 9 \cdot 9 \equiv 13 \pmod{17}$$

$$x_{2G} = s^2 - 2x_G \qquad x_{2G} \equiv 13^2 - 2(5) \equiv 16 - 10 \equiv 6 \pmod{17}$$

$$y_{2G} = s(x_G - x_{2G}) - y_G \qquad y_{2G} \equiv 13(5-6) - 1 \equiv -13 - 1 \equiv -14 \equiv 3 \pmod{17}$$

$$2G = (6,3)$$

Illustration of computing : 3G

3G = 2G + G

P=2G=(6,3) , Q=G=(5,1)

s= (1-3)/ (5-6)= (-2/-1)= (2) mod 17

➔ s= 2

X_R= 2*2 – (6+5) mod 17 = 10

Y_R= 2(6-10) – 3 mod 17 =6

3G = (10, 6)

$$s = \frac{y_P - y_Q}{x_P - x_Q}$$

$$x_R = s^2 - (x_P + x_Q)$$

$$y_R = s(x_P - x_R) - y_P$$

Adding points Using Pre-calculated values

- Use mod n, **n=19**
  - **n: size of the group**
- 3G+4G?
  - (3+4) mod **19** = 7
  - 3G+4G = 7G = (0,6)
- 14G+ 16G
  - (14+16) mod **19** = 11
  - 14G + 16 G = 11G = (13,10)

$$E: \; y^2 \equiv x^3 + 2x + 2 \pmod{17}$$

$G = (5,1)$	$11G = (13,10)$
$2G = (6,3)$	$12G = (0,11)$
$3G = (10,6)$	$13G = (16,4)$
$4G = (3,1)$	$14G = (9,1)$
$5G = (9,16)$	$15G = (3,16)$
$6G = (16,13)$	$16G = (10,11)$
$7G = (0,6)$	$17G = (6,14)$
$8G = (13,7)$	$18G = (5,16)$
$9G = (7,6)$	$19G = \mathcal{O}$
$10G = (7,11)$	

$n = 19$

$h = 1$

3) Elliptic Curves over GF(2^m)

- Use cubic equation where:
  - Variables and coefficients take on values in GF(2^m)
  - Calculations are performed using the rules of arithmetic in GF(2^m).
- Cubic equation appropriate for cryptographic for GF(2^m) is slightly different than for Z_p
  - $y^2 + \mathbf{xy} = x^3 + ax^2 + b$
- $E_{2^m}(a, b)$consists of all pairs of integers (x, y) that satisfy above equation, in addition to $\mathcal{O}$ (the point at infinity or the zero point).

Computing Points on the Elliptic Curve over $GF(2^m)$

The following are the steps to Compute Elliptic Curve points over $E(2^m)$:

- Define the valid values of x's and y's in $E(2^m)$:
  - Select a irreducible polynomial over $GF(2^m)$
  - Select a generator **g**
  - Compute powers of **g**, which are the points in $E(2^m)$
- Compute G and its group: G, 2G, 3G, … , nG
  - Select Elliptic curve equation
  - Compute the points pairs (other than O) that that satisfies this elliptic equations. The points of the pairs are from $E(2^m)$.
  - Select Generator G, and generate the group

Elliptic Curves over GF(2^m): Example

- Assume finite field GF(2^4) with the irreducible polynomial $f(x) = x^4 + x + 1$
- The field has a generator ***g*** that satisfies f(***g***) = 0
  - $f(g) = g^4 + g + 1$ ➔
  - $g = g^4 + 1$
    - $x^4 + 1 \mod (x^4 + x + 1) \equiv x$ ➔ in binary, g = 0010
  - g^2: $g \times g \equiv x \times x \equiv x^2$ ➔ in binary, g^2 = 0100
  - g^3: $g^2 \times g = x^2 \times x \equiv x^3$ ➔ in binary, g^3 = 1000
- Higher exponent of ***g*** is calculated easier by math manipulation. For example:
  - g^4 : $f(g) = g^4 + g + 1 = 0$, ➔ $g^4 = g + 1$ ➔ in binary, g^4 = 0011
  - $g^5 = (g^4)(g) = (g + 1)(g) = g^2 + g$ = 0110.
  - Rest of values are below table.

When g values are less than x^4,
Use x multiplication

$g^0 = 0001$	$g^4 = 0011$	$g^8 = 0101$	$g^{12} = 1111$
$g^1 = 0010$	$g^5 = 0110$	$g^9 = 1010$	$g^{13} = 1101$
$g^2 = 0100$	$g^6 = 1100$	$g^{10} = 0111$	$g^{14} = 1001$
$g^3 = 1000$	$g^7 = 1011$	$g^{11} = 1110$	$g^{15} = 0001$

Elliptic Curves over GF(2^m): Example

- Assume elliptic curve equation: **$y^2 + xy = x^3 + g^4x^2 + 1$** ; where: **$a = g^4$** , **$b = g^0 = 1$**
- Next, we compute the point pairs that satisfies.

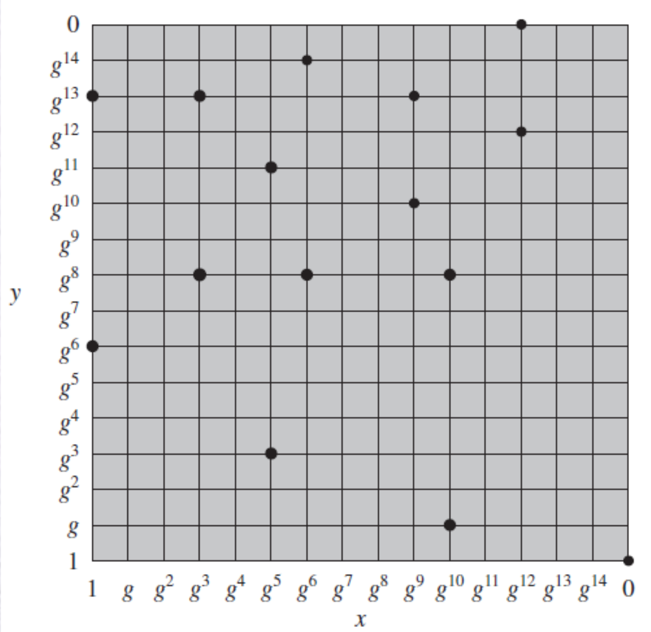
One point that satisfies this equation is (g^5, g^3):

$$(g^3)^2 + (g^5)(g^3) = (g^5)^3 + (g^4)(g^5)^2 + 1$$
$$g^6 + g^8 = g^{15} + g^{14} + 1$$
$$1100 + 0101 = 0001 + 1001 + 0001$$
$$1001 = 1001$$

- The following table lists the points (other than O) that are part of E_{2^4} (g^4, 1). The Figure plots the points of E_{2^4} (g^4, 1).

Table 10.2 Points (other than O) on the Elliptic Curve $E_{2^4}(g^4, 1)$

$(0, 1)$	(g^5, g^3)	(g^9, g^{13})
$(1, g^6)$	(g^5, g^{11})	(g^{10}, g)
$(1, g^{13})$	(g^6, g^8)	(g^{10}, g^8)
(g^3, g^8)	(g^6, g^{14})	$(g^{12}, 0)$
(g^3, g^{13})	(g^9, g^{10})	(g^{12}, g^{12})



Rules of ECC Addition for Abelian Group E_{2^m}

It can be shown that a finite abelian group can be defined based on the set $E_{2^m}(a, b)$, provided that $b \neq 0$. The rules for addition can be stated as follows. For all points $P, Q \in E_{2^m}(a, b)$:

1. $P + O = P$.
2. If $P = (x_P, y_P)$, then $P + (x_P, x_P + y_P) = O$. The point $(x_P, x_P + y_P)$ is the negative of P, which is denoted as $-P$.
3. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ with $P \neq -Q$ and $P \neq Q$, then $R = P + Q = (x_R, y_R)$ is determined by the following rules:

$$x_R = \lambda^2 + \lambda + x_P + x_Q + a$$
$$y_R = \lambda(x_P + x_R) + x_R + y_P$$

where

$$\lambda = \frac{y_Q + y_P}{x_Q + x_P}$$

4. If $P = (x_P, y_P)$ then $R = 2P = (x_R, y_R)$ is determined by the following rules:

$$x_R = \lambda^2 + \lambda + a$$
$$y_R = x_P^2 + (\lambda + 1)x_R$$

where

$$\lambda = x_P + \frac{y_P}{x_P}$$

Elliptic Curve Cryptography (ECC)

- Elliptic Curve is similar to Diffie Hellmann Algorithm.

	Deffien Hellmann	**ECC : $E_q(a,b)$**
Global Public Parameters	q: prime number α: primate root of q	$\{p, a, b, G, n, h\}$ p : field (modulo p) a, b : curve parameters G : Generator Point n : ord(G) (n is the size of the group) h : cofactor
Operation	Multiplication	“dot”
Private Keys	X_A (Alice) , X_B (Bob)	n_A (Alice) , n_B (Bob)
Public Keys	$Y_A = \alpha^{XA}$ (Alice) $Y_B = \alpha^{XB}$ (Bob)	$P_A = n_A G$ (Alice) $P_B = n_B G$ (Bob)
Final Shared Key by Bob/Alice	Key=$\alpha^{X_B X_A}$ mod q	K=$n_A n_B G$

Key Exchange & Encryption

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
$K = n_B \times P_A$

Encryption/Decryption

To encrypt P_m from A to B:

- **A** chooses a random positive integer ***k***
- Cipher text is:

$$C_m = \{C_1, C_2\} = \{kG, P_m + kP_B\}$$

To decrypt, B:

- multiply first point by Bob private Key: $kG \times n_B$
- **Subtract from second point:**

$$C_2 - n_B C1$$
$$= P_m + kP_B - n_B(kG)$$
$$= P_m + k(\cancel{n_B G}) - n_B(\cancel{kG})$$
$$= P_m$$

Example: Key Exchange

- $p = 211$
- $E_p(0,-4) \rightarrow y^2 = x^3 - 4$
- $G = (2, 2)$

- $n_A = 121$
- $\rightarrow P_A = 121(2, 2) = (115, 48)$

- $n_B = 203$
- $\rightarrow P_B = 203(2, 2) = (130, 203)$

$K = n_A \times P_B = 121(130, 203) = (161, 69)$

$K = n_B \times P_A = 203(115, 48) = (161, 69)$

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

$K = n_B \times P_A$

Another Example

(notice the difference in notation)

Bob	Attacker	Alice
Bob picks	$y^2 \equiv x^3 + 2x + 2 \pmod{17}$	Alice picks
$\beta = 9$	$G = (5,1)$	$\alpha = 3$
Computes	$n = 19$	Computes
$B = 9G = (7,6)$	$A = (10,6)$	$A = 3G = (10,6)$
Receives	$B = (7,6)$	Receives
$A = (10,6)$		$B = (7,6)$
Computes		Computes
$\beta A = 9A = 9(3G) = 27G = 8G = (13,7)$		$\alpha B = 3B = 3(9G) = 27G = 8G = (13,7)$

27 mod (n=19) $\equiv$ 8

Example: Encryption

$q = 257$;
$E_q(a, b) = E_{257}(0, -4)$
$y^2 = x^3 - 4$;
$G\ (2, 2)$
$P_m = (112, 26)$

Bob:
Bob private key : $n_B = 101$
Bob pubic key: $P_B = n_B \times G = 101(2, 2) = (197, 167)$

$P_B = (197, 167)$

<u>Alice:</u>
Alice picks private: $k = 41$
$C_1 = k \times G = 41(2, 2) = (136, 128)$
$k \times P_B = 41(197, 167) = (68, 84)$
$P_m + k \times P_B = (112, 26) + (68, 84) = (246, 174)$
$C_m = (C_1, C_2) = \{(136, 128),(246, 174)\}$

$C_m = (C_1, C_2) = \{(136, 128),(246, 174)\}$

<u>Bob</u> computes
$C_2 - n_B \times C_1$
$= (246, 174) - 101(136, 128)$
$= (246, 174) - (68, 84)$
$= (112, 26)$

Encryption/Decryption

To encrypt P_m from A to B:
- **A** chooses a random positive integer ***k***
- Cipher text is:

$C_m = \{kG, P_m + kP_B\}$

To decrypt, B:
- multiply first point by Bob private Key: $kG \times n_B$
- **Subtract from second point:**

$Pm + kPB - nB(kG)$
$= Pm + k(\cancel{nBG}) - nB(\cancel{kG})$
$= Pm$

Security of Elliptic Curve Cryptography

- Depends on the difficulty of the elliptic curve logarithm problem
- Fastest known technique is “Pollard rho method”
- Compared to factoring, can use much smaller key sizes than with RSA
- For equivalent key lengths computations are roughly equivalent
- Hence, for similar security ECC offers significant computational advantages

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