ELLIPTIC CURVE CRYPTOGRAPHY

SLIDES BY HARKEERAT BEDI

Public Key Cryptography

- Components
  - Public Key
  - Private Key
  - Set of Operators that work on these Keys
  - Predefined Constraints (required by some algorithms)

Elliptic Curve Cryptography

- Components

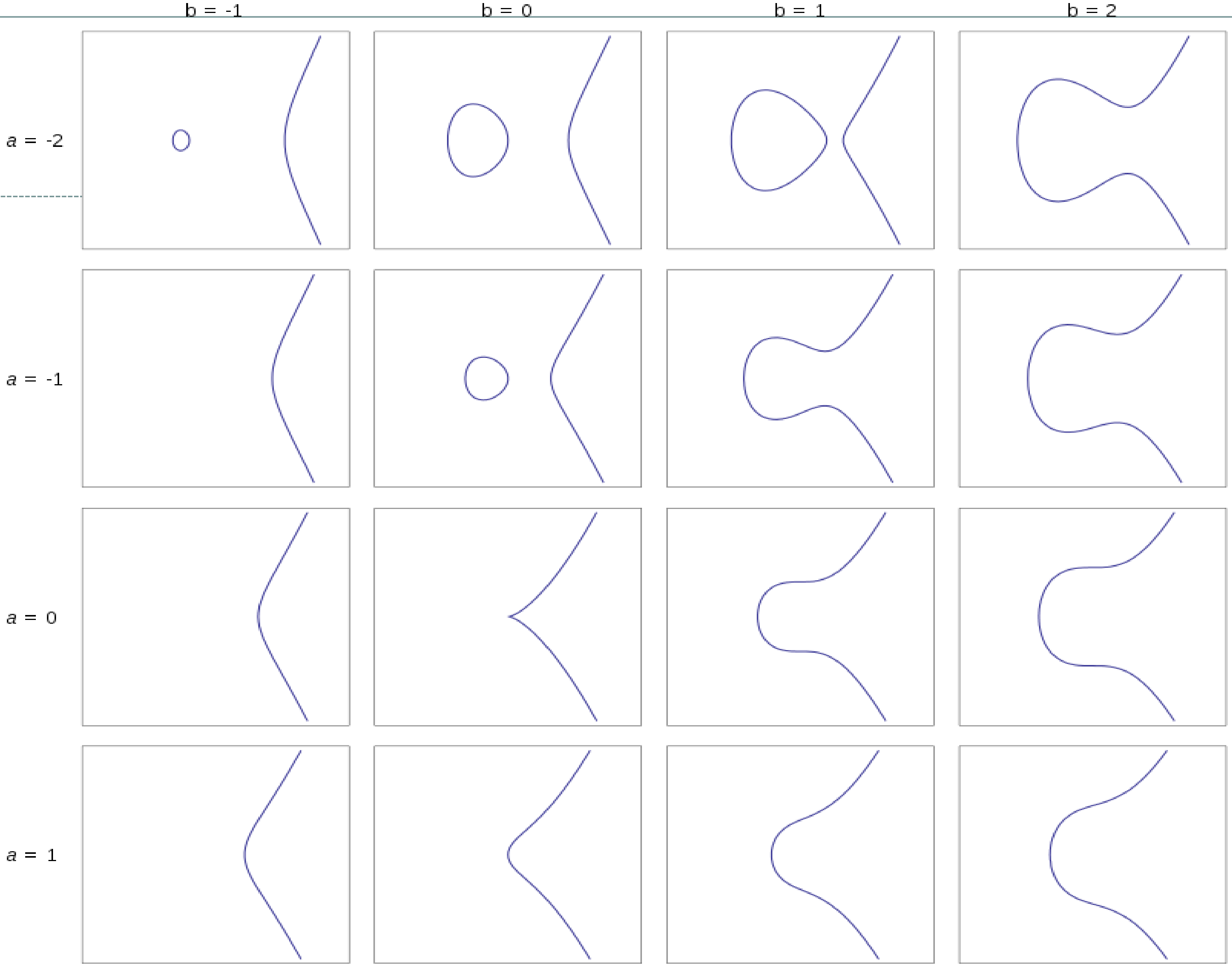
Private Key	**Public Key**	**Set of Operations**	**Domain Parameters (Predefined constants)**
A random number	Point on a curve = Private Key * G	These are defined over the curve $\mathbf{y^2 = x^3 + ax + b,}$ where $\mathbf{4a^3 + 27b^2 \neq 0}$	G, a, b

General form of a EC

- An *elliptic curve* is a plane curve defined by an equation of the form

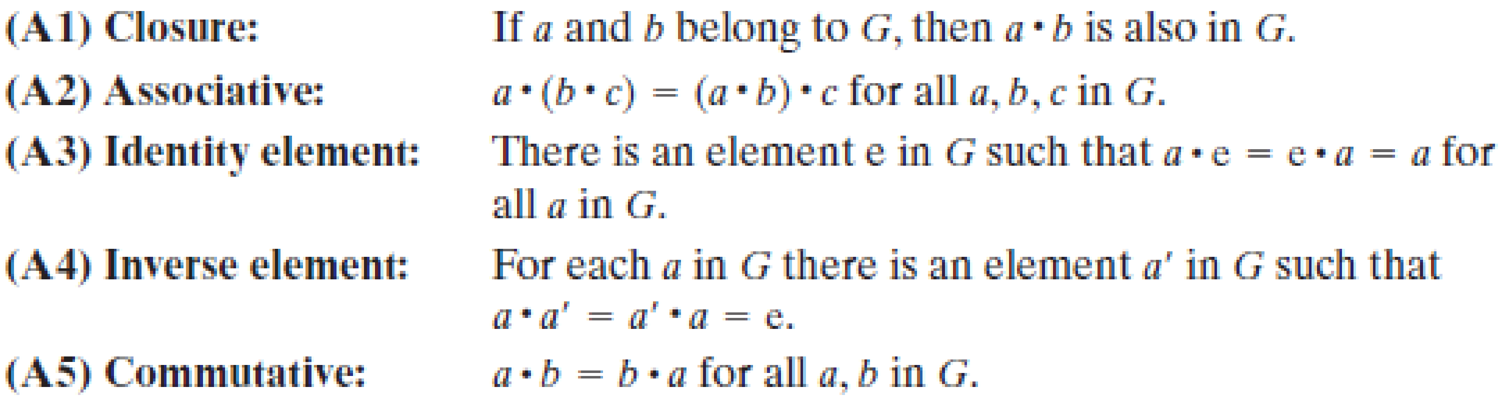
$$y^2 = x^3 + ax + b$$

where $\mathbf{4a^3 + 27b^2 \neq 0}$


b = -1
b = 0
b = 1
b = 2
a = -2
a = -1
a = 0
a = 1

Characteristics of Elliptic Curve

- Forms an abelian group
  - Symmetric about the x-axis
  - *Point at Infinity* acting as the identity element



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

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

(A3) Identity element: There is an element e in G such that $a \cdot \mathrm{e} = \mathrm{e} \cdot a = a$ for all a in G.

(A4) Inverse element: For each a in G there is an element a' in G such that $a \cdot a' = a' \cdot a = \mathrm{e}$.

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

Discrete Logarithm Problem (DLP)

- Let P and Q be two points on the elliptic curve
  - Such that Q = kP, where k is a scalar value
- DLP: Given P and Q, find k?
  - If k is very large, it becomes computationally infeasible
- The security of ECC depends on the difficulty of DLP
- Main operation in ECC is Point Multiplication

Point Multiplication

- Point Multiplication is achieved by two basic curve operations:

1. Point Addition, L = J + K
2. Point Doubling, L = 2J

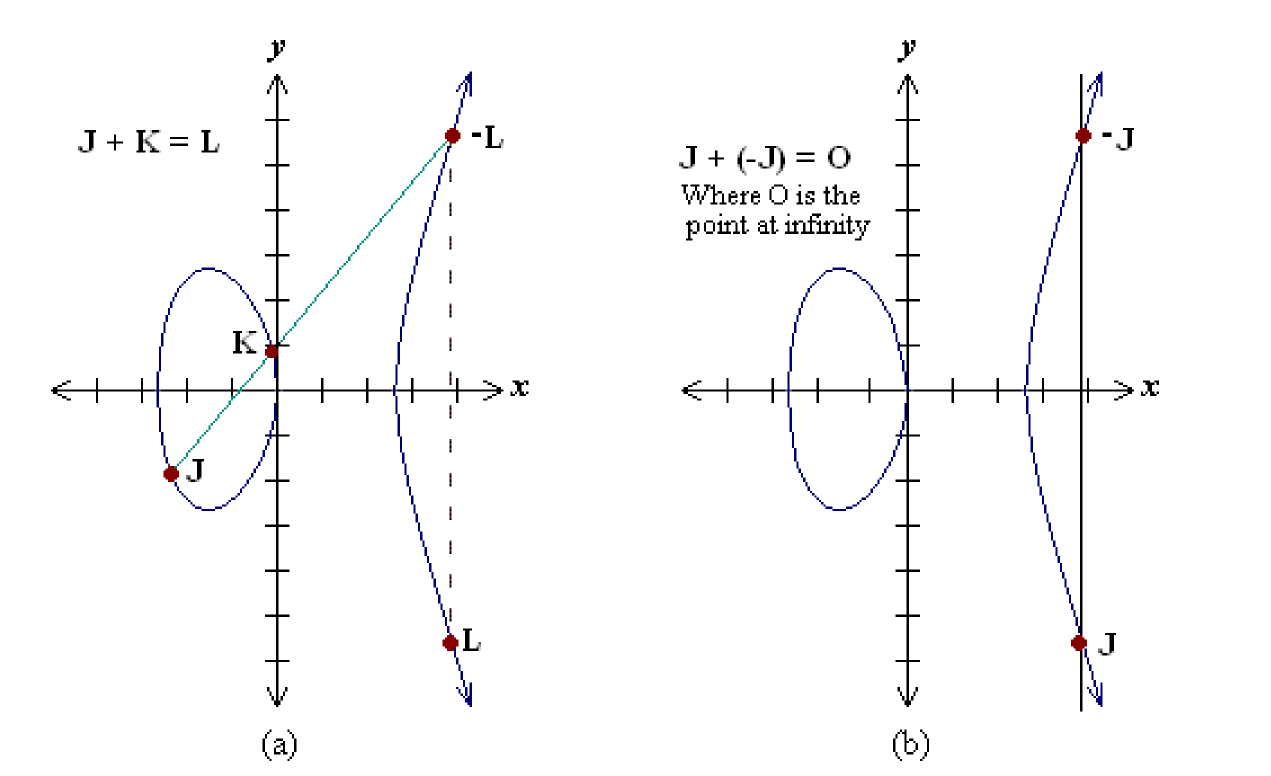
Example:

If k = 23;   then, kP = 23*P

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

Point Addition

Geometrical explanation:



Point Addition

Analytical explanation:

- Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)
- Let L = J + K where L = (x_L, y_L), then

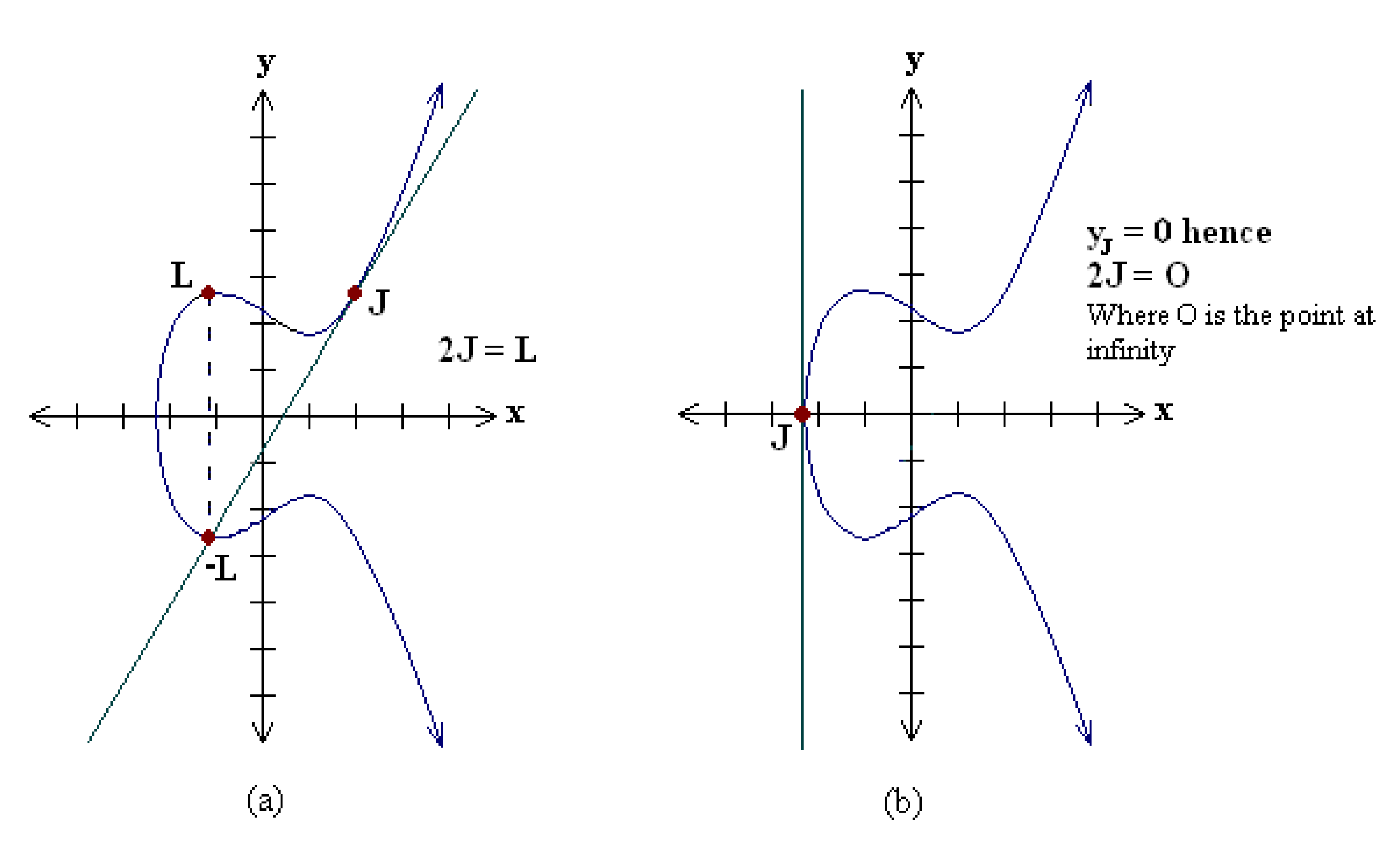
$$x_L = s^2 - x_J – x_K$$

$$y_L = -y_J + s\ (x_J – x_L)$$

$s = (y_J – y_K)/(x_J – x_K)$, s is slope of the line through J and K.

Point Doubling

Geometrical explanation:



Point Doubling

Analytical explanation

- Consider a point J such that $J = (x_J, y_J)$, where $y_J \neq 0$
- Let $L = 2J$ where $L = (x_L, y_L)$, Then

$$x_L = s^2 - 2x_J$$

$$y_L = -y_J + s(x_J - x_L)$$

$s = (3x_J^2 + a) / (2y_J)$, s is the tangent at point J and a is one of the parameters chosen with the elliptic curve.

Finite Fields

- The Elliptic curve operations shown were on real numbers
  - Issue: operations are slow and inaccurate due to round-off errors

- To make operations more efficient and accurate, the curve is defined over two finite fields

  1. Prime field GF(p) and

  2. Binary field GF($_2{}^m$)

- The field is chosen with finitely large number of points suited for cryptographic operations

EC on Prime field GF(p)

- Elliptic Curve equation:

  $$y^2 \bmod p = x^3 + ax + b \bmod p$$

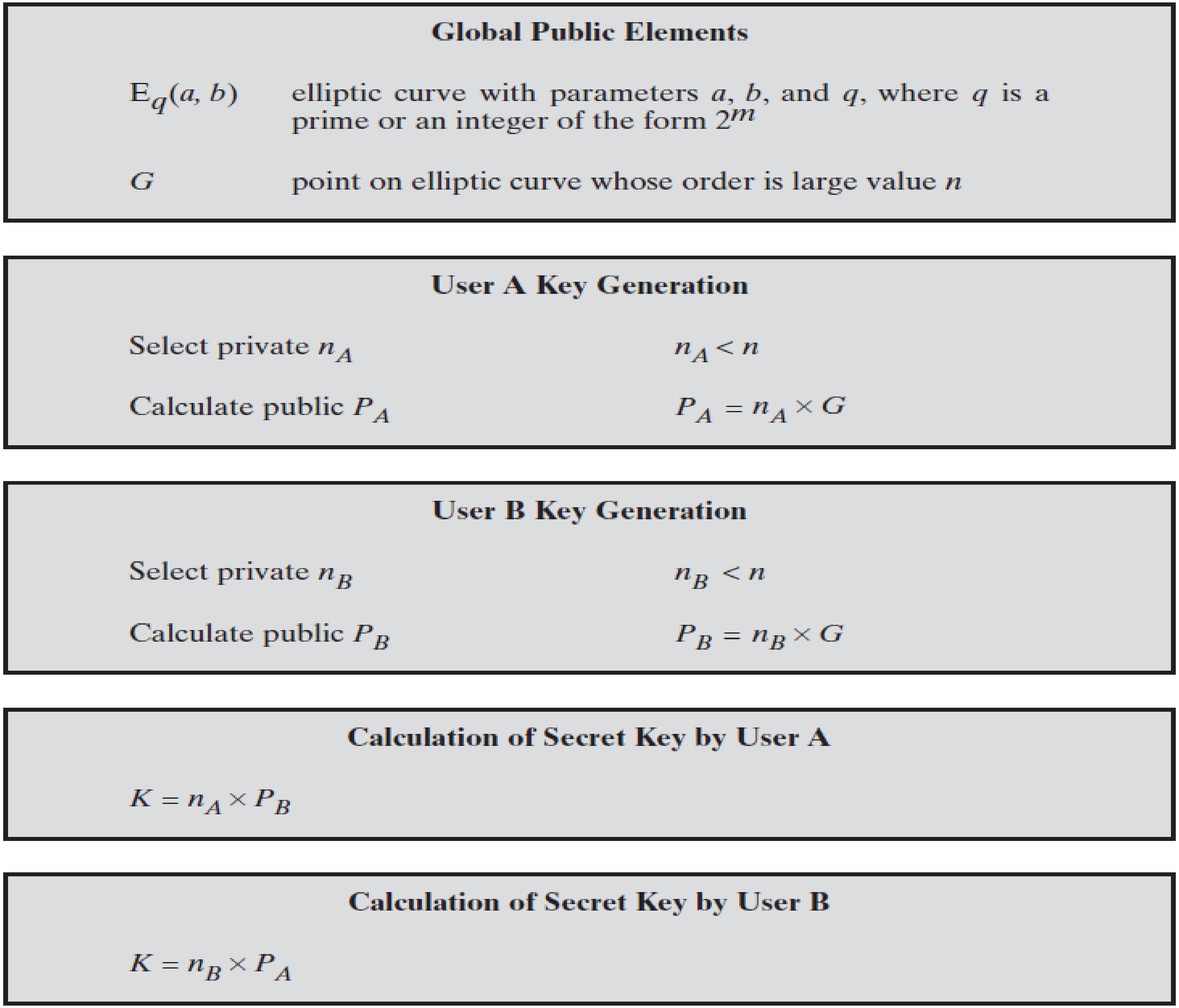
  where $4a^3 + 27b^2 \bmod p \neq 0$.
- Elements of finite fields are integers between 0 and p-1
- The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.
- SEC (Standard for Efficient Cryptography) specifies curves with p ranging between 112-521 bits

EC on Binary field GF($_2{}^m$)

- Elliptic Curve equation:

  $$\mathbf{y^2 + xy = x^3 + ax^2 + b,}$$

  where **b ≠ 0**
- Here the elements of the finite field are integers of length at most **m** bits.
- In binary polynomial the coefficients can only be 0 or 1.
- The **m** is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.
- SEC specifies curves with **m** ranging between 113-571 bits



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

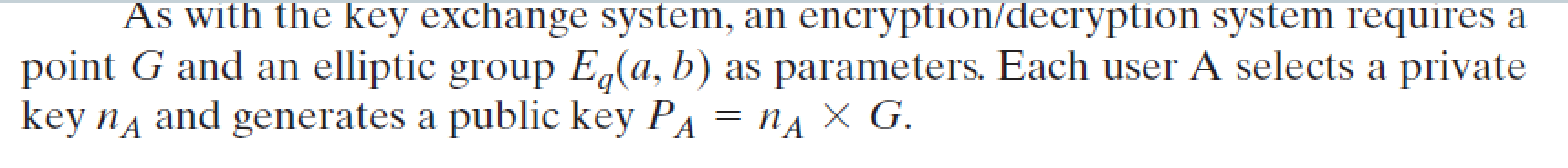
$K = n_A \times P_B$

Calculation of Secret Key by User B

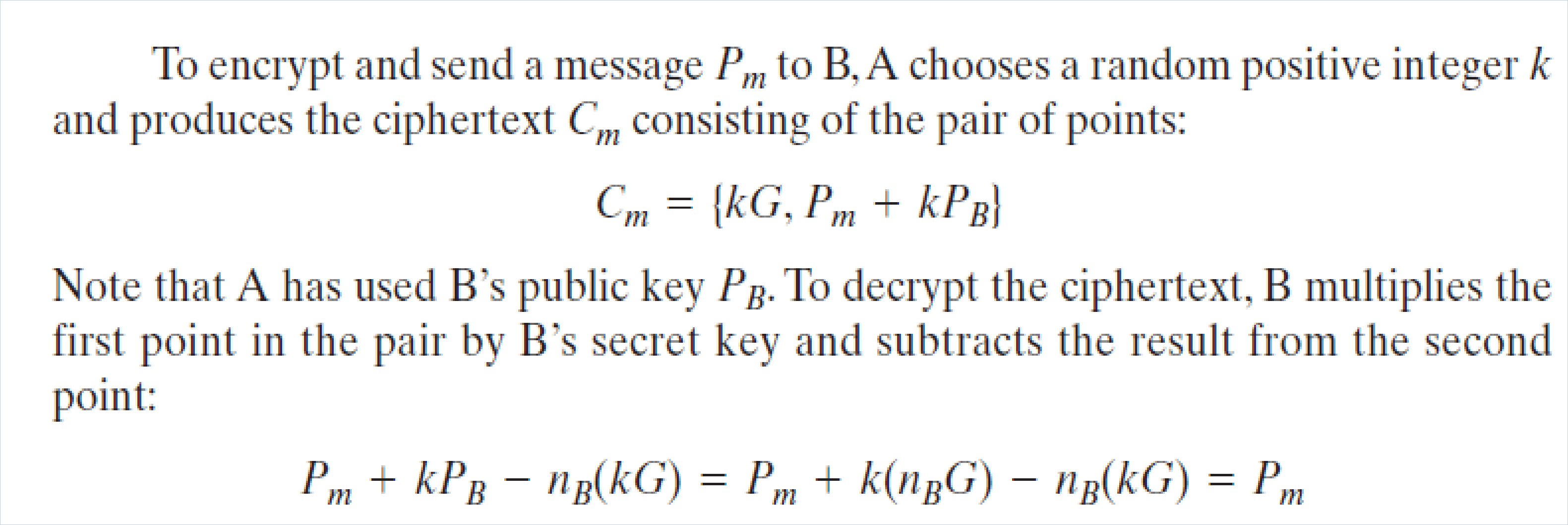
$K = n_B \times P_A$

Figure 10.7 ECC Diffie-Hellman Key Exchange

Elliptic Curve Encryption/Decryption



As with the key exchange system, an encryption/decryption system requires a point G and an elliptic group $E_q(a, b)$ as parameters. Each user A selects a private key n_A and generates a public key $P_A = n_A \times G$.



To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext C_m consisting of the pair of points:

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

Note that A has used B's public key P_B. To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

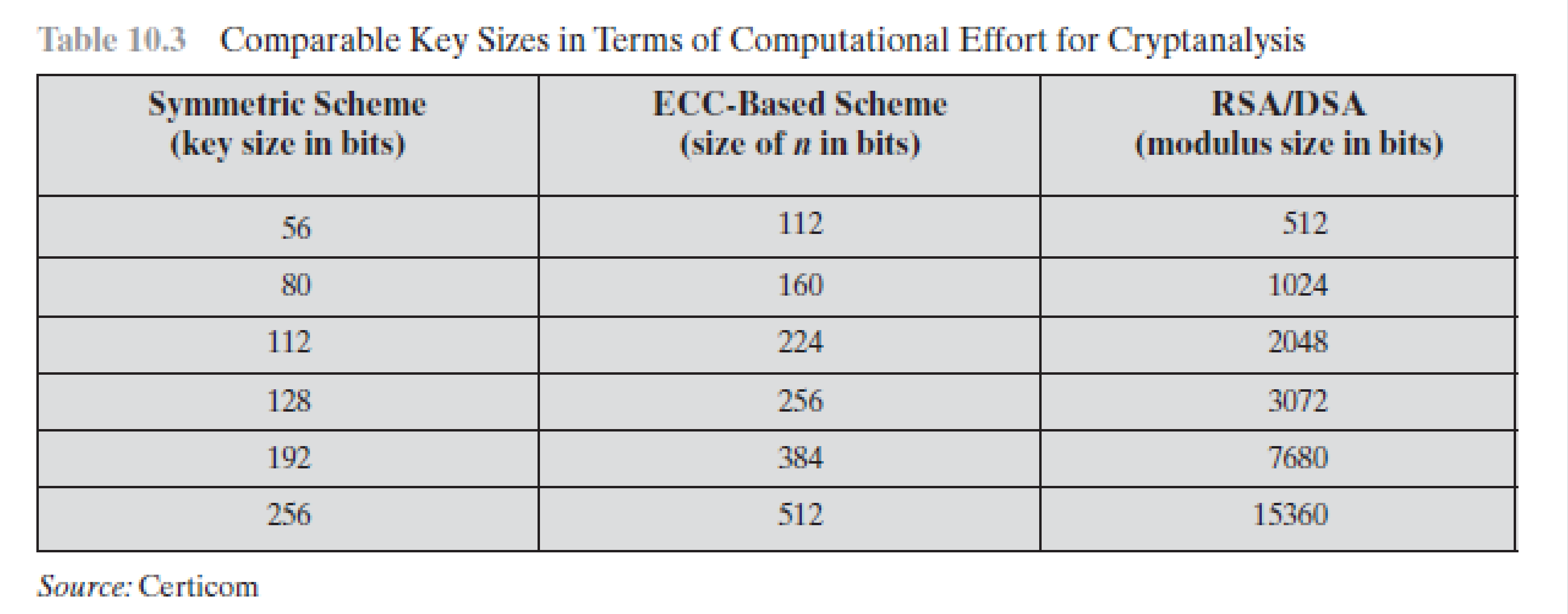


Table 10.3 Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis

Symmetric Scheme (key size in bits)	**ECC-Based Scheme (size of *n* in bits)**	**RSA/DSA (modulus size in bits)**
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Source: Certicom