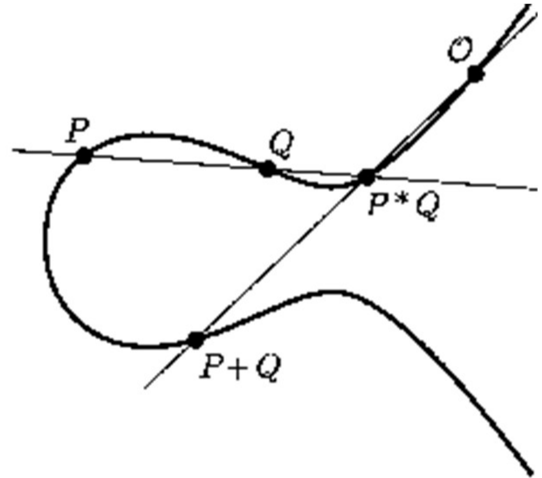


Elliptic Curve Cryptography (ECC)

Jibi Abraham

COEP Tech

COEP TECHNOLOGICAL UNIVERSITY

Shivajinagar, Pune-411 005

(A Unitary Technological University of Govt. of Maharashtra)

How do we analyze Cryptosystems?

- How difficult is the underlying problem that it is based upon
  - RSA – Integer Factorization
  - DH – Discrete Logarithms
  - ECC - Elliptic Curve Discrete Logarithm problem
  - How do we measure difficulty?
    - We examine the algorithms used to solve these problems

RSA and ECC

- Computational overhead of the RSA-based approach to PKC increases with the size of the keys
- As algorithms for integer factorization have become more and more efficient, the RSA based methods have had to resort to longer and longer keys
- Elliptic curve cryptography (ECC) can provide the same level and type of security as RSA, but with much shorter keys
- ECC takes one-sixth the computational effort to provide the same level of cryptographic security that you get with 1024-bit RSA

Comparison: Symmetric Encryption, RSA and ECC

- Best current estimates of the key sizes for three different approaches to encryption for comparable levels of security against brute-force attacks
- "brute-force" for AES means searching through the entire key-space, integer factorization for RSA and solving the discrete logarithm for ECC

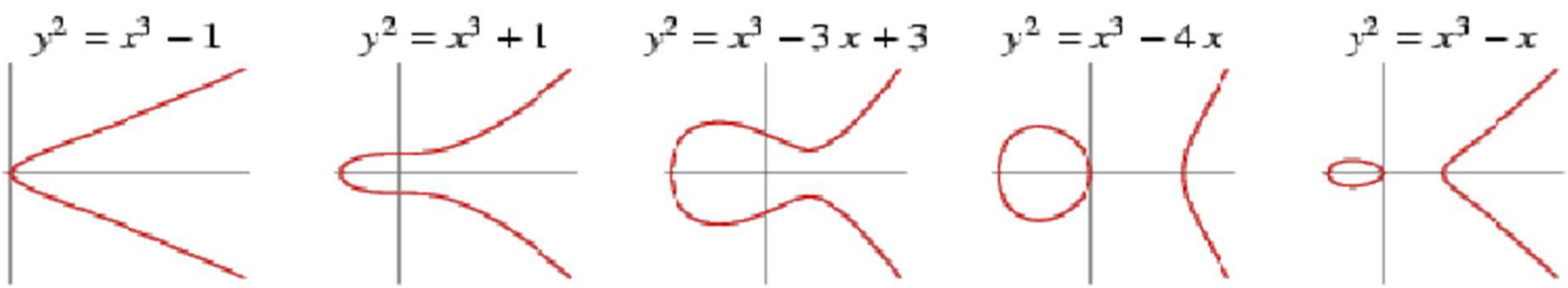
Symmetric Encryption Key Size in bits	*RSA and Diffie-Hellman "Key" size in bits*	*ECC "Key" Size in bits*
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Applications of ECC

- Many devices are small and have limited storage and computational power
- Where can we apply ECC?
  - **Wireless communication devices**
  - Smart cards
  - Web servers that need to handle many encryption sessions
  - **Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems**

ECC for Low Resource Devices

- Because of the much smaller key sizes involved, ECC algorithms can be implemented on smartcards without mathematical coprocessors
- Contactless smart cards work only with ECC because other systems require too much induction energy
- Since shorter key lengths translate into faster handshaking protocols, ECC is also becoming increasingly important for wireless communications.



Main Idea of ECC

- Imagine a set of points (x_i , y_i) in a plane denoted by E. Set is very very large, but finite
- Can define a group operator on E denoted by the symbol '+'
- Given two points P and Q $\in$ E, group operator calculate a third point R $\in$ E, such that P + Q = R
- Given a point G $\in$ E, we are interested in using the group operator to find G + G, G + G + G, G + G + G + . . . + G for an arbitrary number of repeated invocations of the group operator

ECC in Nutshell

- Given an ordinary integer k, use the notation $k \times G$ to represent the repeated addition $G + G + . . . + G$ in which G makes k appearances, with the operator '+' being invoked $k - 1$ times
- Set E is magical in the sense that, after calculated $k \times G$ for a given $G \in E$, it is extremely difficult to recover k from $k \times G$
- All of the assumptions we made above are satisfied when the set E of points (x_i , y_i) is drawn from an elliptic curve
- To find k, finding a discrete logarithm is hard

Elliptic Curves

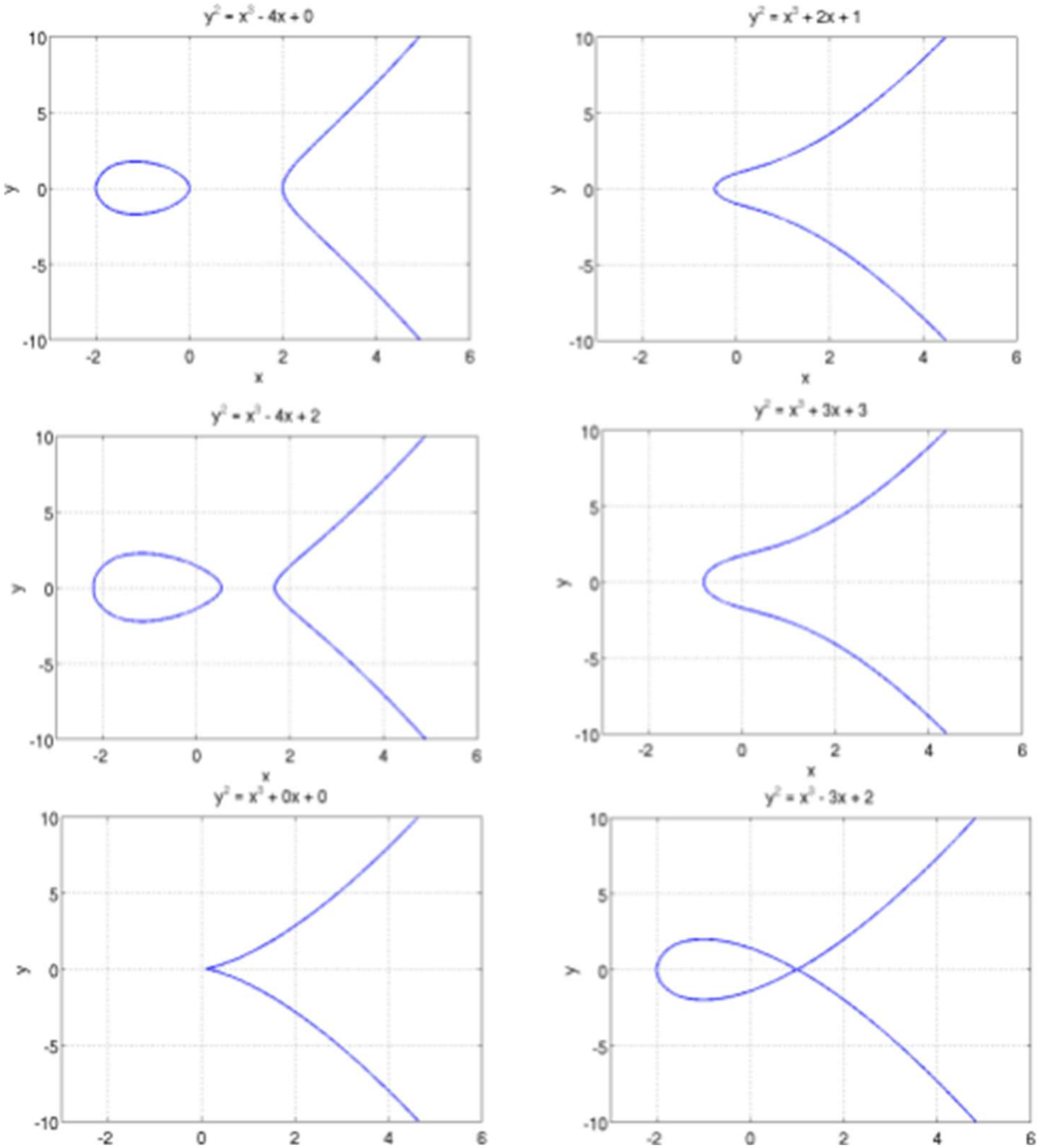
- Elliptic curves have nothing to do with ellipses. Ellipses are formed by quadratic curves. Elliptic curves are always cubic
- Simplest possible "curves" are, straight **lines**
- The next simplest possible curves **are conics, being quadratic** as $ax^2 + bxy + cy^2 + dx + ey + f = 0$
  - If $b^2 - 4ac < 0$, curve is either an ellipse or circle or point, or the curve does not exist;
  - if it is equal to 0, a parabola, or two parallel lines, or no curve at all;
  - if it is greater than 0, a hyperbola or two intersecting lines.

Elliptic Curves

- The next simplest possible curves are **elliptic curves**
- An elliptic curve has its form $y^2 = x^3+ax+b$ for some fixed values for the parameters a and b
- Values of x, y, a, and b are drawn from a set that must at least be a ring with a multiplicative identity element
- An elliptic curve used in the Microsoft Windows Media Digital Rights Management Version 2:

$$y^2 = x^3 + 317689081251325503476317476413827693272746955927x + 79052896607878758718120572025718535432100651934$$

Elliptic Curves with different a and b



Elliptic Curves for Cryptography

- Cryptography requires error-free arithmetic
- Two families of elliptic curves are used in cryptographic applications: prime curves over Z_p and binary curves over $GF(2^n)$
- Prime curves over Z_p are best for software applications, because the extended bit-fiddling operations needed by binary curves are not required;
- Binary curves over $GF(2^n)$ are best for hardware applications, where it takes remarkably few logic gates to create a powerful, fast cryptosystem

Prime Elliptic Curves

- By restricting the values of a, b, x, and y to some prime finite field Z_p (in the set of integers from 0 through p-1 and in which calculations are performed modulo p)
- Elliptic curves appropriate for cryptography over Z_p, is described by $y^2 \equiv (x^3 + ax + b) \pmod{p}$ ----- (1)

subject to a constraint $(4a^3 + 27b^2) \neq 0 \pmod{p}$ ---- (2)

Prime Elliptic Curves - Example

- $y^2 \equiv (x^3 + ax + b) \pmod p$ ----- (1)

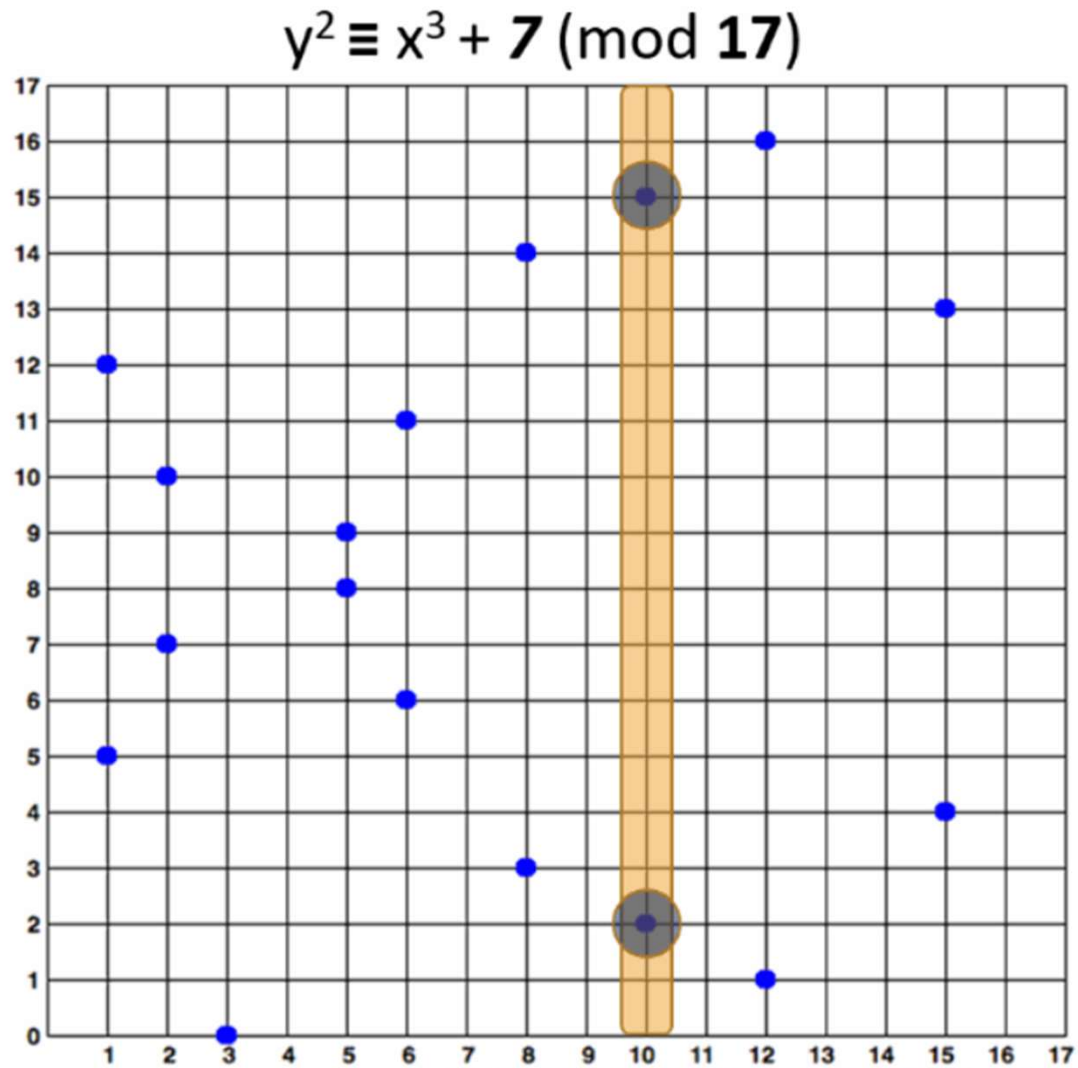
  subject to a constraint $(4a^3 + 27b^2) \neq 0 \pmod p$ ---- (2)
- Example: a = 1, b = 1, p = 23, what is the Elliptic Curve?
  - $4+27 \bmod 23 \neq 0 \Rightarrow y^2 \equiv (x^3+x+1) \bmod 23$ is an ECC
- Is x = 9, y = 7, a point on this curve?
  - $7^2 \bmod 23 = 3$
  - $(9^3+9+1) \bmod 23 = 739 \bmod 23 = 3$
  - (9,7) is a point on this curve

Points on the EC

- $E_p(a, b)$ represents all the points (x, y) that obey the 2 conditions: $y^2 \equiv (x^3 + ax + b) \pmod p$ and $(4a^3 + 27b^2) \neq 0 \pmod p$
- The set of points in $E_p(a, b)$ is no longer a curve, but a collection of discrete points in the (x, y) plane defined by the Cartesian product $Zp \times Zp$
- Elliptic curves over finite fields $\mathbb{F}$**p** (in the Weierstrass form) have **at most 2 points per y coordinate** (odd **x** and even **x**).

Points on the EC

- Elliptic curves over finite fields $\mathbb{F}$**p** (in the Weierstrass form) have **at most 2 points per y coordinate** (odd **x** and even **x**).

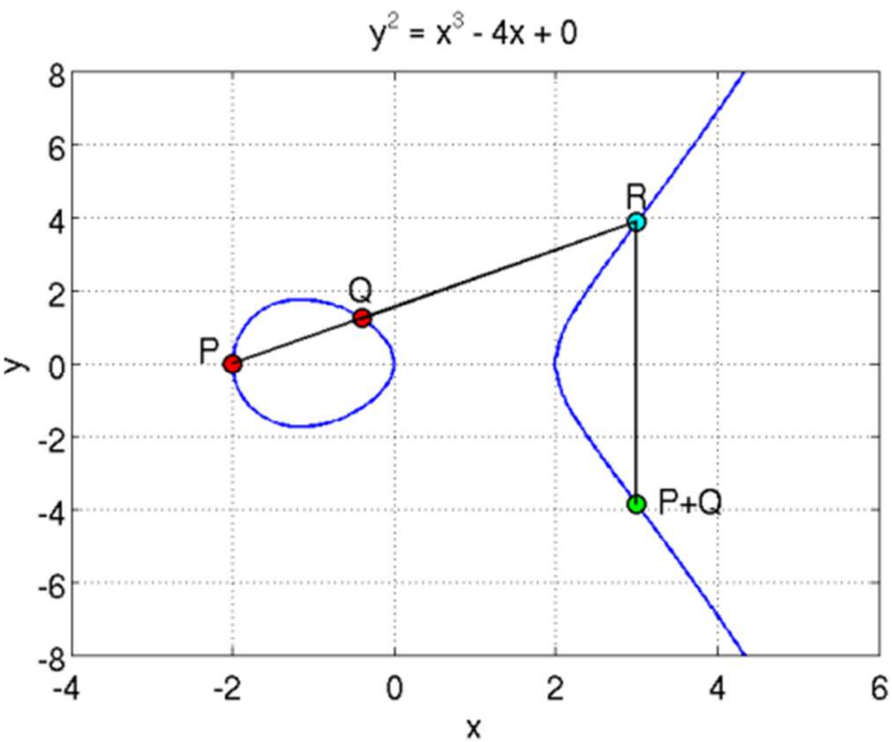


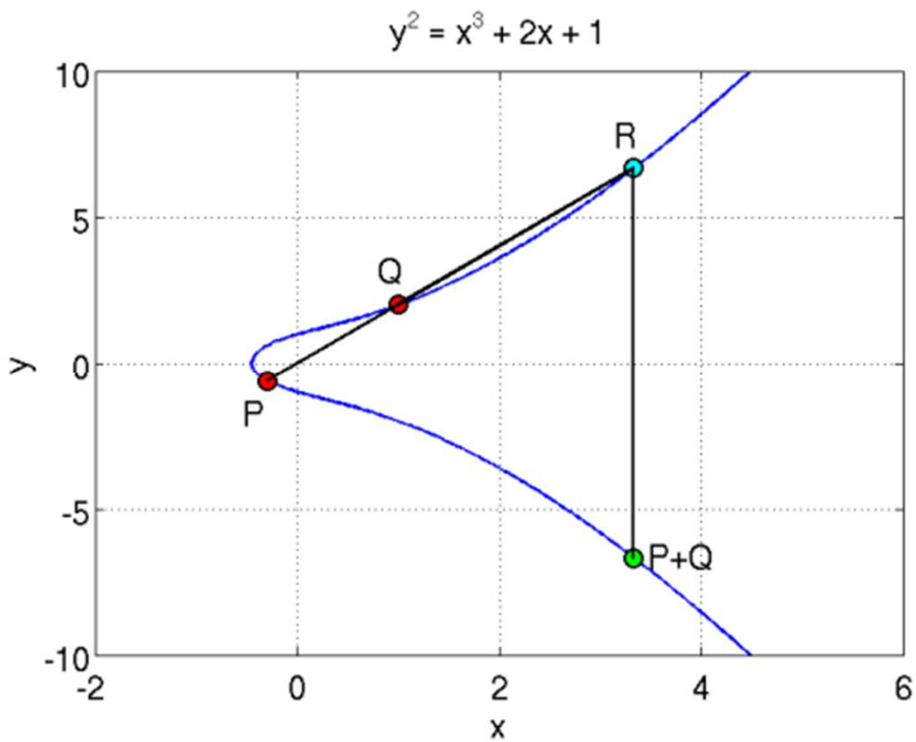
Points on the EC

- Given a point P, one cannot show geometrically how to compute 2P, or given two points P and Q, one cannot show geometrically how to determine P + Q
- The algebraic expressions derived for P+Q and 2P etc. continue to hold good provided the calculations are carried out modulo p

Addition on EC

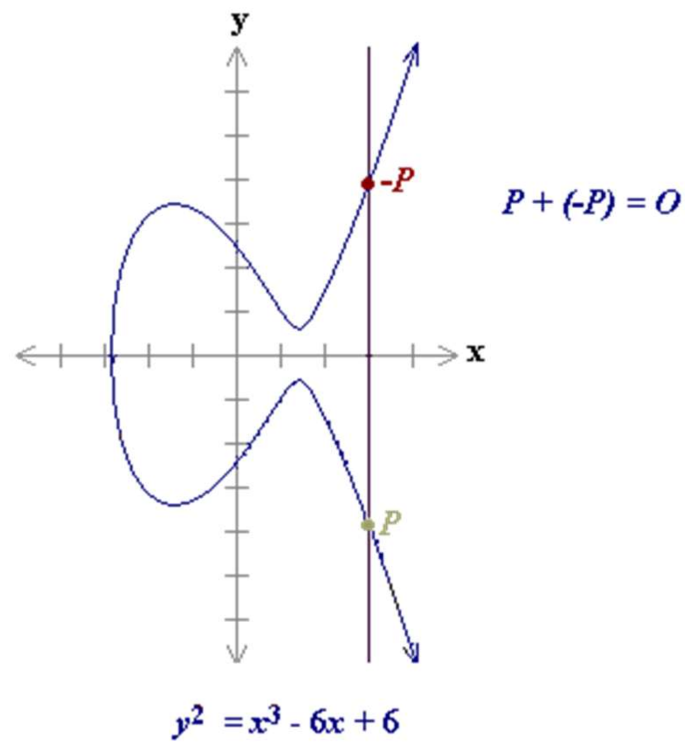
- To add a point P on an elliptic curve to another point Q on the same curve, join P with Q with a straight line
- Third point R is the intersection of this straight line with the curve.
- If R exists, the mirror image of this point with respect to the x-coordinate is the point P + Q





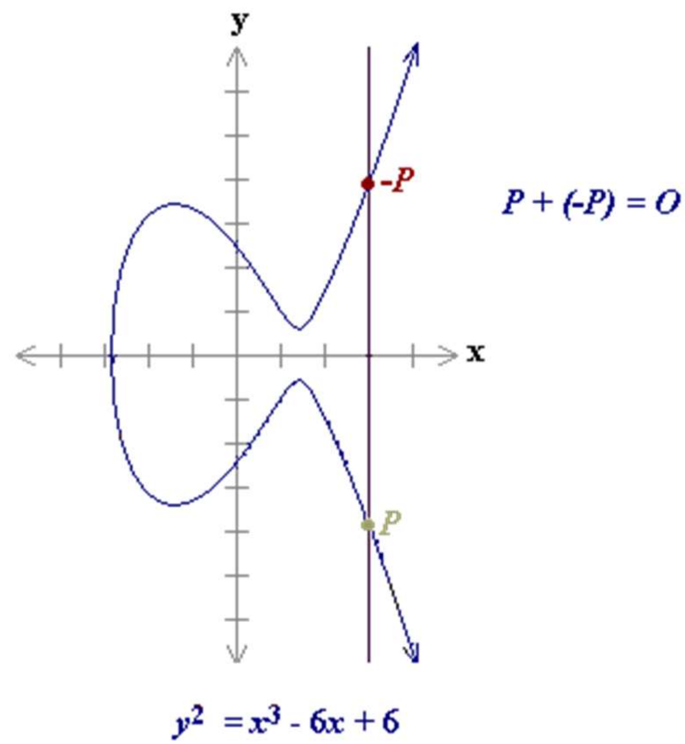
Point at Infinity O

- If the intersection of P and Q does not exist, we say it is at infinity.
- This infinity is at the distinguished point O, whose mirror reflection is also at O. Therefore, for such points, $P + Q = O$ and $Q = -P$
- The point represented by O is actually at infinity — along the y-axis.



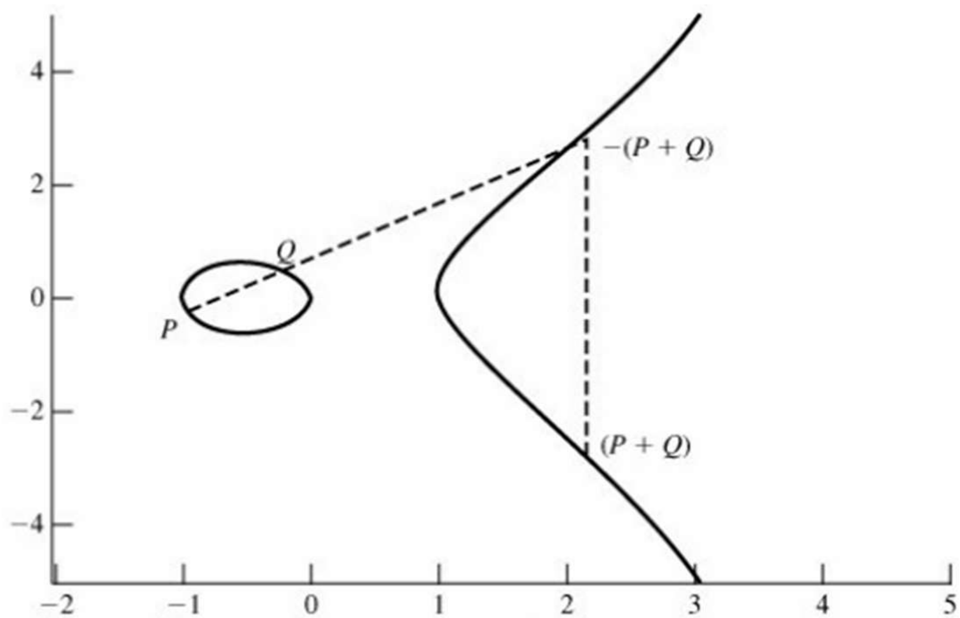
Additive Inverse on EC

- O serves as the additive identity element for the group
- We stipulate that P + O = P for any point on the curve
- Will further stipulate that that O + O = O, implying that −O = O
- ***O* is called the additive identity of the elliptic curve group.**
- Hence all elliptic curves have an additive identity ***O***.



Additive Inverse on EC

- The negative of a point P is the point with the same x coordinate but the negative of the y coordinate; that is, if P = (x, y), then -P = (x, -y). Then P + (-P) = O
  - These two points can be joined by a vertical line.
- The only time when the line joining P and Q does NOT intersect the curve is when that line is parallel to the y-axis
- We define the additive inverse of a point P as its mirror reflection with respect to the x axis



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

Example: Inverse of a Point

- Consider EC with a = b = 1, p=23, $y^2 = x^3 + x + 1$
- Points 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)

- Consider a point P = (13,7) in $E_{23}(1,1)$, find -P
- -P = (13, -7). But -7 mod 23 = 16.
- -P = (13, 16), which is also in E23(1,1).

Algebraic Addition of 2 Points

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

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

Addition Example

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

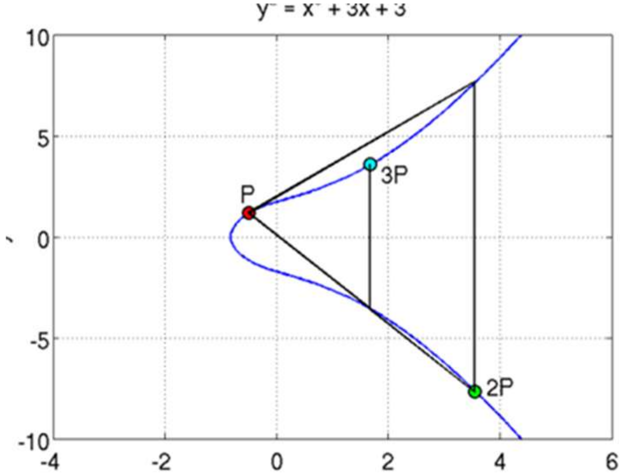
$$\lambda = \begin{cases} \left(\dfrac{y_Q - y_P}{x_Q - x_P}\right) \bmod p & \text{if } P \neq Q \\ \left(\dfrac{3x_P^2 + a}{2y_P}\right) \bmod p & \text{if } P = Q \end{cases}$$

- Eg: P = (3,10) and Q = (9,7) in $E_{23}(1,1)$, find P+Q

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

- $2^{-1} \bmod 23 = 12$, $-12 \bmod 23 = 11$
- $x_R = (11^2 – 3 – 9) \bmod 23 = 17$ $y_R = (11(3 –17) –10) \bmod 23 = 164 \bmod 23 = 20$, P + Q = (17, 20)

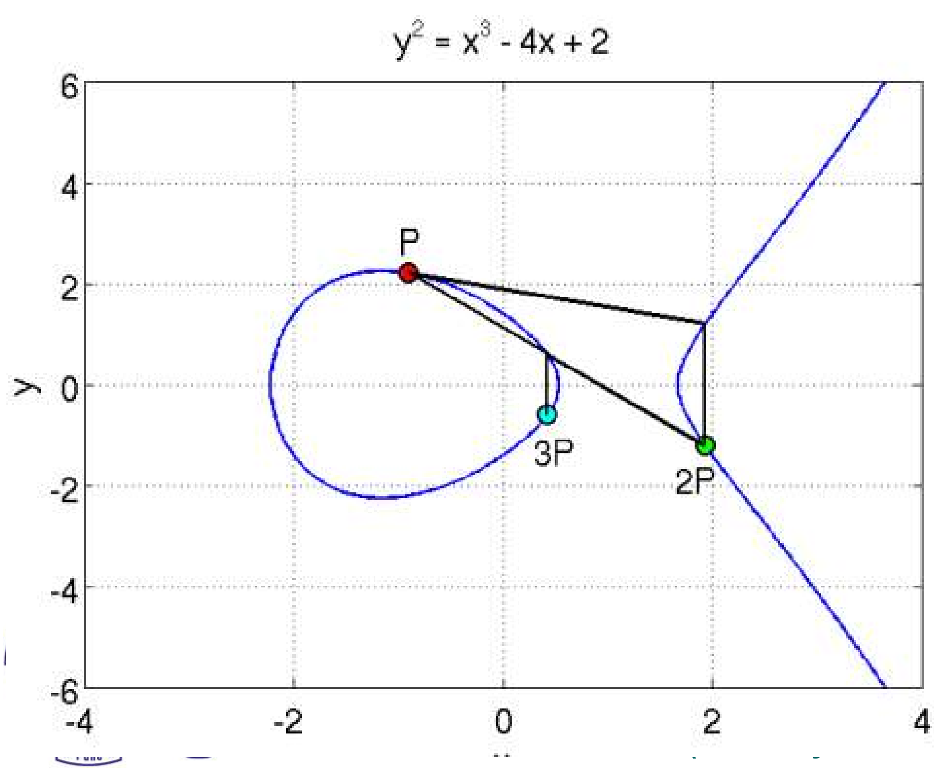
Multiplication on EC

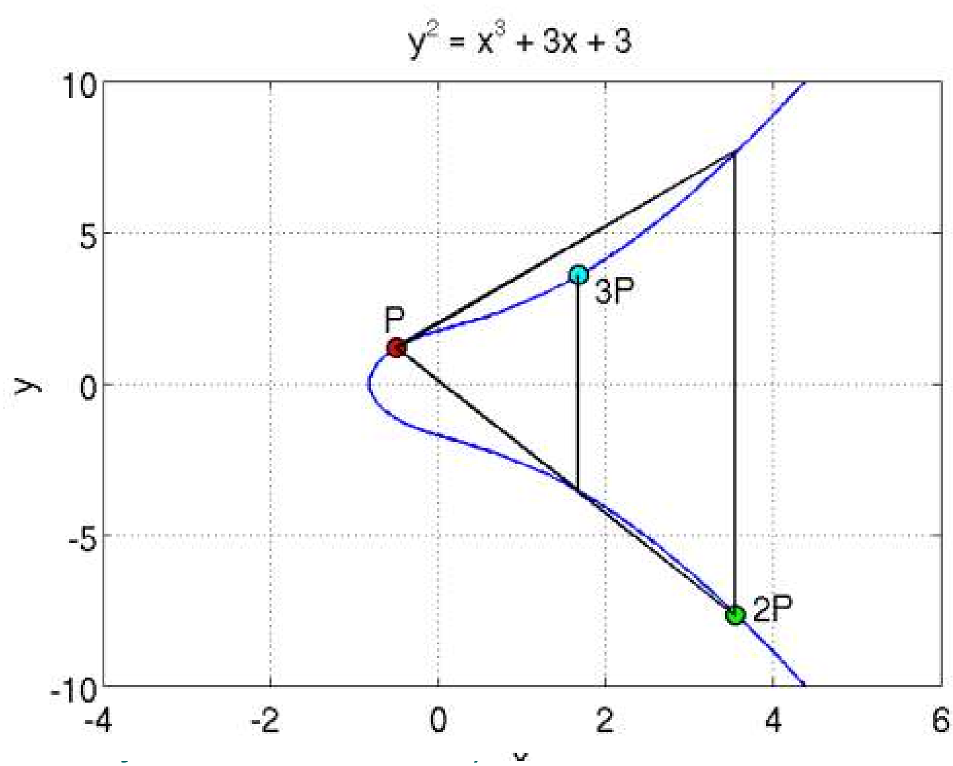


- What is the additive inverse of a point where the tangent is parallel to the y-axis?
- The additive inverse of such a point is the point itself.
- If the tangent at P is parallel to the y-axis, then P + P = O
- Consider two distinct points P and Q and let Q approach P
- Line joining P and Q will obviously become a tangent at P in the limit
- The operation P + P means that we must draw a tangent at P, find the intersection of the tangent with the curve, and then take the mirror reflection of the intersection
- To double a point Q, draw the tangent line and find the other point of intersection

Multiplication on EC

- Can express P + P as 2P, P + P + P as 3P, and so on
- Can define "multiplication" as repeated addition.
- Therefore, k × P = P + P + . . . + P with P making k appearances on the right





Multiplication Example

- $x_R = (\lambda^2 - x_P - x_Q) \bmod p$,
- $y_R = (\lambda(x_P - x_R) - y_P) \bmod p$
- Example: $P = (3,10)$ in $E_{23}(1,1)$. To find 2P

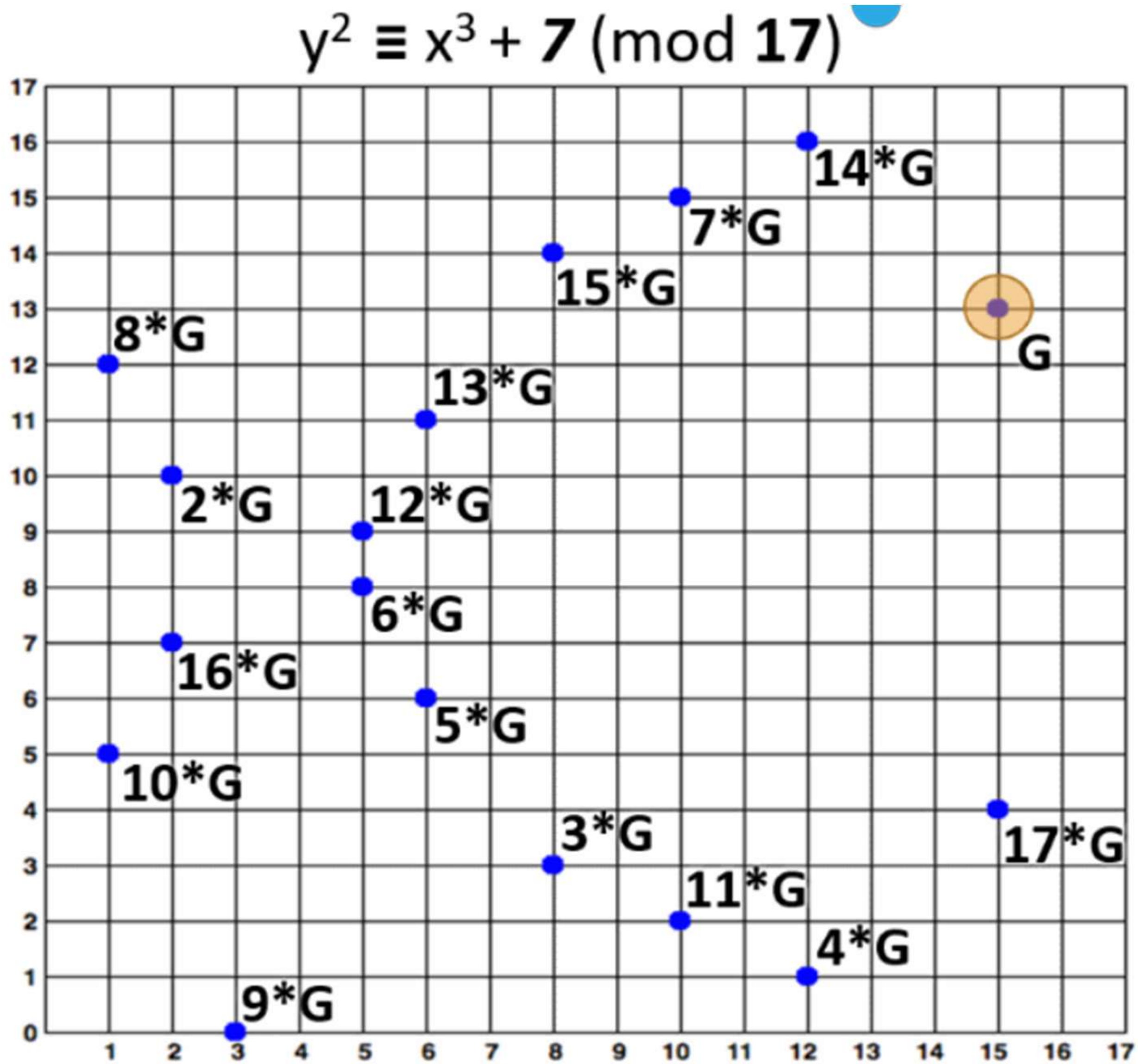
$$\lambda = \left(\frac{3x_P^2 + a}{2y_P}\right) \bmod p$$

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

- $x_R = (6^2 - 2.3) \bmod 23 = 30 \bmod 23 = 7$
- $y_R = (6(3 - 7) - 10) \bmod 23 = (-34) \bmod 23 = 12$
- $2P = (7, 12)$

ECC Multiplication

- EC points, generated by multiplying the generator point **G** {15, 13} by 2, 3, 4, ..., 17



ERSITY

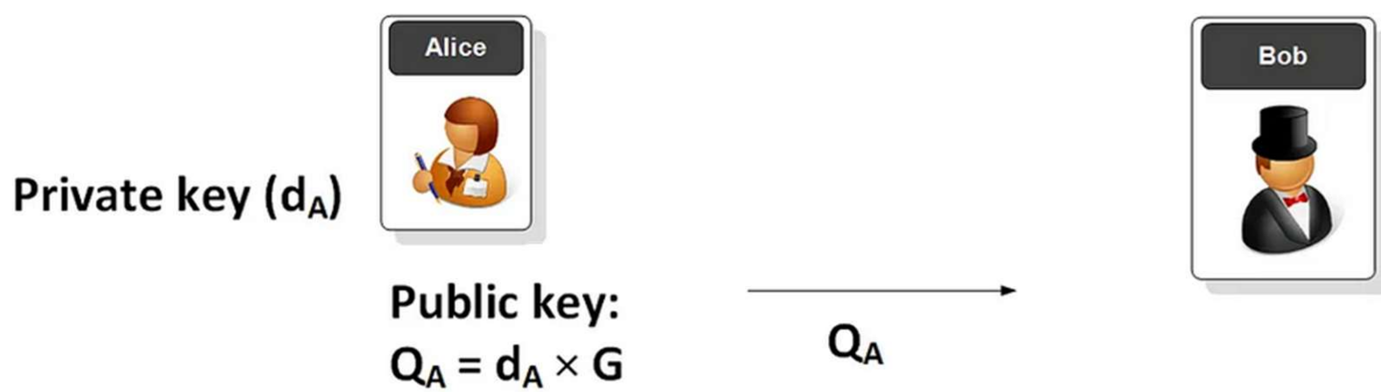
(A Unitary Technological University of Govt. of Maharashtra)

Encryption in ECC

- The addition operation in ECC is the counterpart of modular multiplication in RSA and multiple addition is the counterpart of modular exponentiation.
- Consider the equation $Q = kP$ where $Q, P \in E_p(a, b)$ and $k < p$.
- The discrete logarithm problem for elliptic curves
  – With a proper choice for P, it is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P.

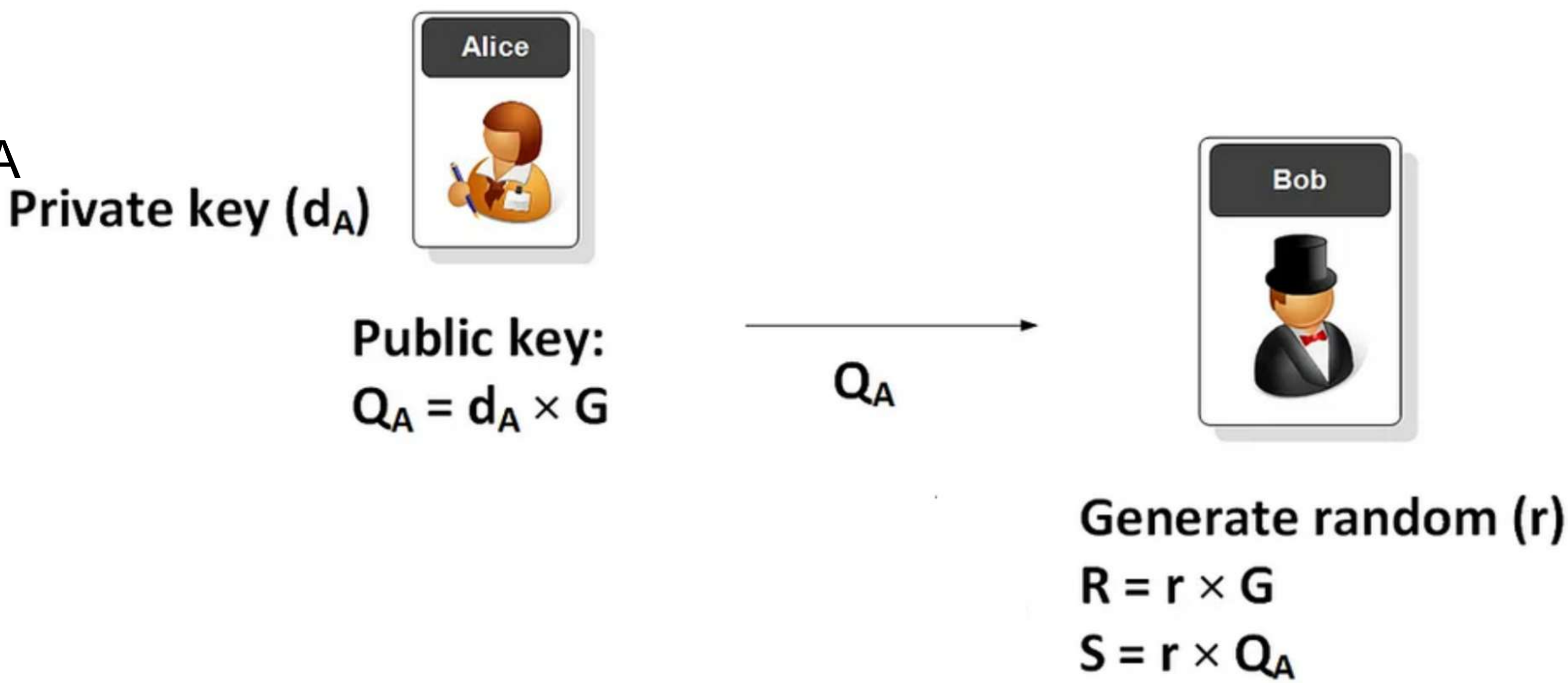
Key Generation for Encrypt/Decrypt

- Chooses the parameters p, a, and b for an elliptic-curve $E_p(a, b)$ with a base point $G \in E_p(a, b)$
- Alice selects the private key an integer d_A
  - Public key is generated by $Q_A = d_A \times G$
  - Keys of Alice: (d_A, Q_A)
- Alice gives Public key Q_A to Bob



Encrypt using ECC

- Alice gives Public key Q_A ($Q_A = d_A \times G$) to Bob
- If Bob has to send a message to Alice, select another random number 'r' to ensure that even for the same message m, the cipher text generated is different each time
- **Encrypt**: Find R and S and share R with Alice
- $R = r \times G$
- $S = r \times Q_A$



COEP Tech

Encrypt using ECC

- Alice receives $R = r \times G$ and finds S

Private key (d_A)

Public key:

$Q_A = d_A \times G$

$\xrightarrow{Q_A}$

Generate random (r)

$R = r \times G$

$S = r \times Q_A$

$\xleftarrow{R}$

$S = d_A \times R$

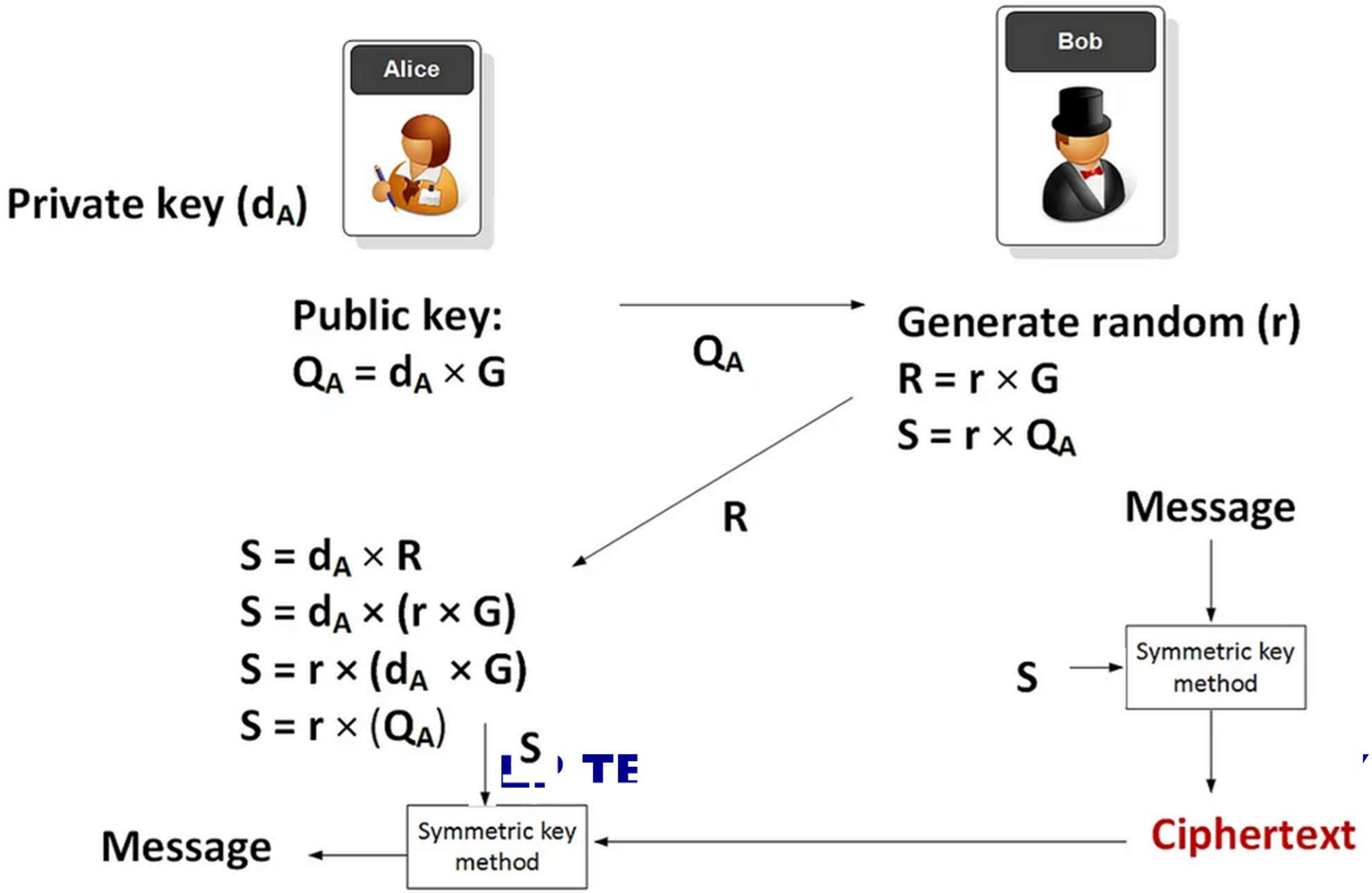
$S = d_A \times (r \times G)$

$S = r \times (d_A \times G)$

$S = r \times (Q_A)$

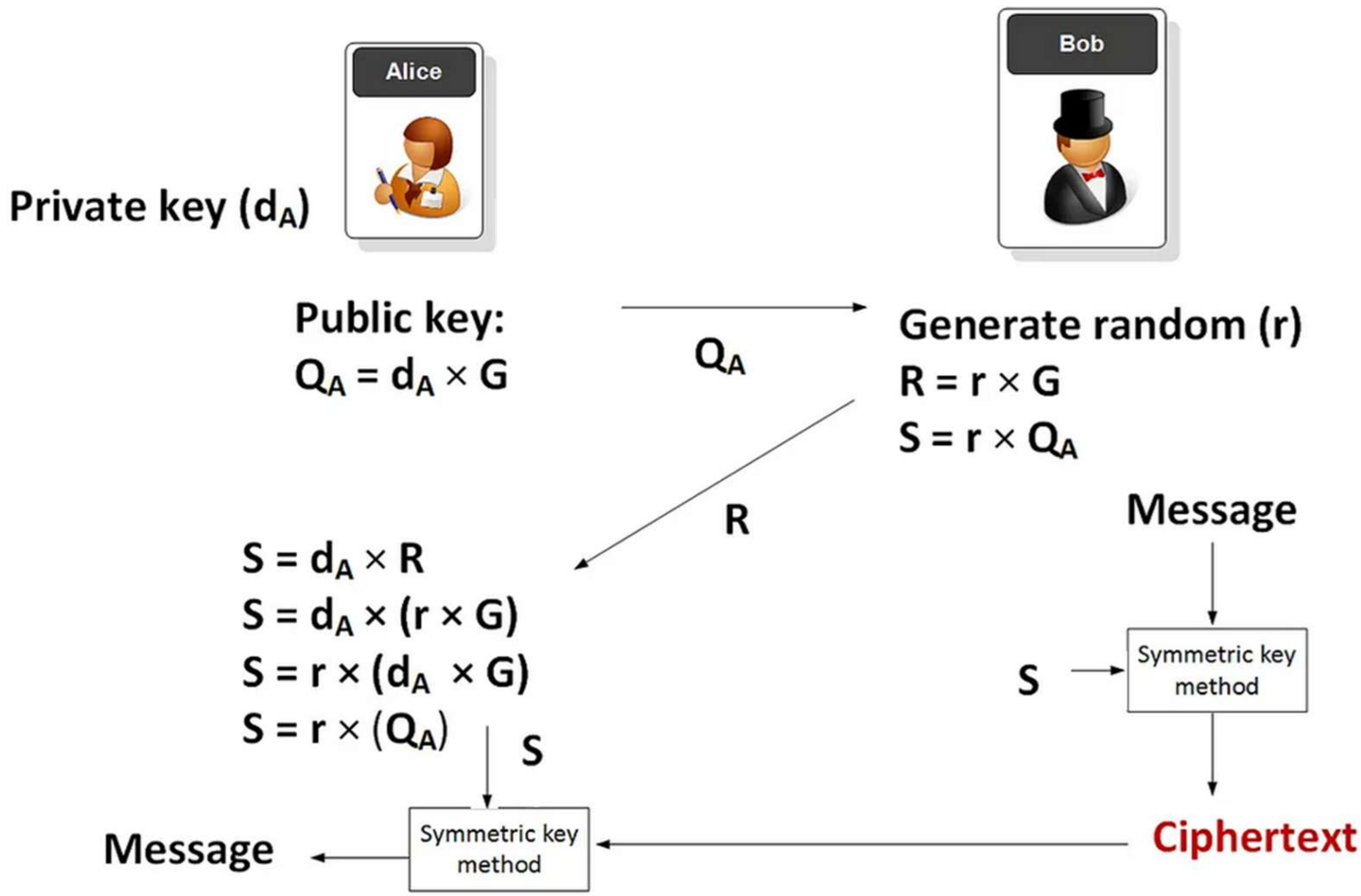
Encrypt using ECC

- If Bob has to send a message m to Alice, m will be converted to a point P_m on the Elliptic Curve



Encrypt/Decrypt using ECC

- Bob encrypt: $C_m = \{P_m \times S\}$
- Alice decrypts $P_m = \{C_m \times S\}$
- Alice then decodes P_m to get the message, M.



Encrypt/Decrypt via ElGamal

- **Encrypt** with Public Key of Alice Q_A

1. Map the message to a point M on the elliptic curve
2. Generate a random integer r
3. Compute $R = r.G$
4. Compute $S = r.Q_A + M$
5. Return the tuple $C = (R, S)$ to Alice

- **Decrypt** the cipher tuple C using the private key, d_A :

1. compute $M = S - R.\ d_A$
   $= r.Q_A + M - r.Q_A$
   $= M$

   Since $R.\ d_A = r.G.\ d_A$
   $= r.\ Q_A$

2. Map the point back to a message

Encrypt/Decrypt Example

- $y^2 = x^3 + x + 6$ over $\mathbb{Z}_{11}$
- generator $G = (2,7)$

$\alpha = (2,7)$	$2\alpha = (5,2)$	$3\alpha = (8,3)$
$4\alpha = (10,2)$	$5\alpha = (3,6)$	$6\alpha = (7,9)$
$7\alpha = (7,2)$	$8\alpha = (3,5)$	$9\alpha = (10,9)$
$10\alpha = (8,8)$	$11\alpha = (5,9)$	$12\alpha = (2,4)$

- choose the private key $d_A = 7$
- Public Key $Q_A = d_A.G = 7.(2,7) = (7,2)$
- Plaintext is $M = (10,9)$, which is a point in E
- Choose a random value for $r = 3$

Encrypt/Decrypt Example

- $R = r.G = 3.(2,7) = (8,3)$
- $S = r.Q_A + M = 3.\ (7,2) + (10,9)$
- $= (10,9) + (3,5) = (10,2)$
- Sends $C = (R, S) = ((8,3),(10,2))$ to other End
- $M = S - R.\ d_A$

  $= (10,2) - 7(8,3)$

  $= (10,2) - (3,5)$

  $= (10,2) + (3,6)$

  $= (10,9)$

Map a message to a point M on EC

- A message can be an arbitrary byte string
- Both byte strings and integers have the same nature
- Split the message into two parts, and interpret the 1st part as an integer x and 2nd part as an integer y.
- But, (x, y) must be a point on the elliptic curve, which may not always be valid
- Solution: Transform the message to x. Then, compute a valid y from the curve equation
- But, issue is that not every x will have a corresponding y
- Luckily, most of the popular elliptic curves used in cryptography have an interesting property: about half of the possible field integers are valid x-coordinates.

Map a message to a point M on EC

- Order is the number of valid points that belong to the elliptic curve group (**256-bit** number)
- Weierstrass curve, for each y, there are 2 possible x-coordinates. So, no of valid x-coordinates is order/ 2
- Example: For a given 384-bit integer, this to be a valid x–coordinate is (order / 2) / (2 ** 384) which is approximately 50%. So, solution is trial and error
- Append a random bytes (padding) to the message
- If the resulting padded message does not translate to a valid x-coordinate, we choose another random padding and try again, until we find one that works

Discrete Logarithm for Elliptic Curves

- An adversary could try to recover M from C = M × P by calculating 2P, 3P, 4P, . . ., kP with k, in the worst case, spanning the size of the set $E_p(a, b)$, and then seeing whether or not the result matched C
- If p is sufficiently large and if the point P on the curve $E_p(a, b)$ is chosen carefully, that would take too much effort
- It is not scalable, attempting to recover M from C by repeated addition would amount to solving an exponentially complex problem with an exhaustive search

Example to Find Discrete Log

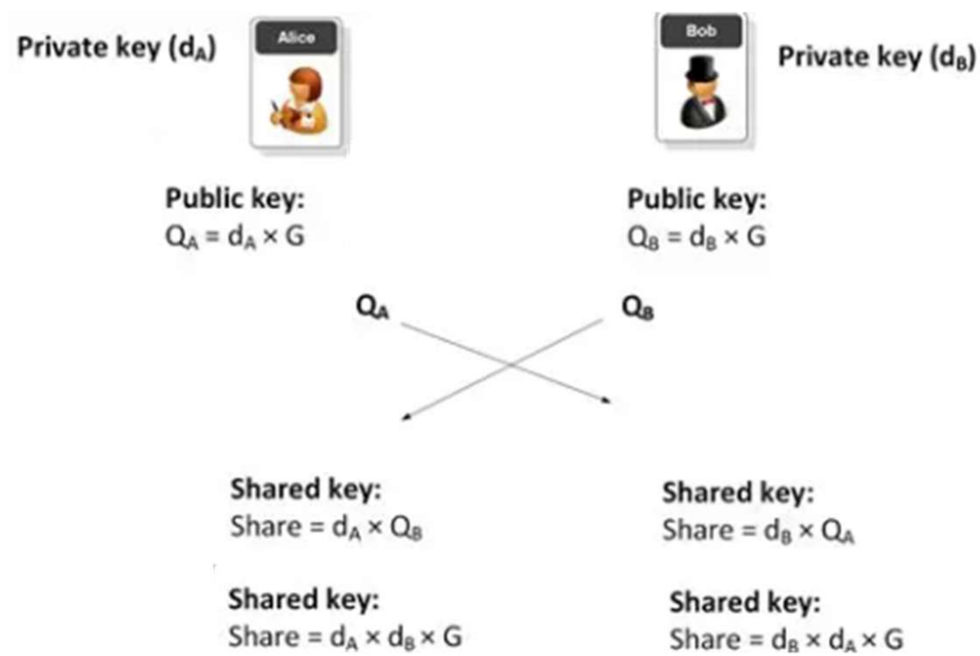
- Consider the group $E_{23}(9, 17)$ defined by the equation $y^2 \bmod 23 = (x^3 + 9x + 17) \bmod 23$.
- Discrete logarithm k of C=(4, 5) to base P=(16, 5)?
- The brute-force method is to compute multiples of P until Q is found.
- P=(16,5); 2P=(20,20); 3P=(14,14); 4P=(19,20); 5P=(13,10);6P=(7,3);7P=(8,7); 8P=(12,17); 9P=(4, 5).
- Because 9P = (4, 5) = C, the discrete logarithm C = (4, 5) to the base P = (16, 5) is k = 9.
- In a real application, k would be so large as to make the brute-force approach infeasible.

Elliptic-Curve Diffie-Hellman (ECDH)

- Chooses the parameters p, a, and b for an elliptic-curve based group $E_p(a, b)$, and a base point $G \in E_p(a, b)$
- A selects private key an integer X_A. Public key is generated by $Q_A = d_A \times G$
- B selects private key an integer X_B. Public key is generated by $Q_B = d_B \times G$

Elliptic-Curve Diffie-Hellman (ECDH)

- A calculates the shared session key by $K = d_A \times Q_B$
- B calculates the shared session key by $K = d_B \times Q_A$
- K as calculated by A = $d_A \times Q_B = d_A \times (d_B \times G) = (d_A \times d_B) \times G = (d_B \times d_A) \times G = d_B \times (d_A \times G) = d_B \times Q_A$ = K as calculated by B



Elliptic Curve over GF

- For a binary curve defined over $GF(2^n)$, the variables and coefficients all take on values in $GF(2^n)$:
  - $y^2 + xy = x^3 + ax^2 + b, b \neq 0$
- Consists of 2^n elements, addition and multiplication operations that can be defined over polynomials
- Example: finite field $GF(2^4)$ with the irreducible polynomial $f(x) = x^4 + x + 1$
  - the elliptic curve $y^2 + xy = x^3 + g^4x^2 + 1$.
  - In this case $a = g^4$ and $b = g^0 = 1$, where g is the generator

Research on Binary ECC Curves

1. Scalar Multiplication: LSB first vs MSB first
   - MSB First: Requires m point doublings and (m-1)/2 point additions on average
   - LSB First: On the average m/2 point Additions and m/2 point doublings
     - Is faster than MSB and can parallelize
2. Montgomery Technique of Scalar Multiplication: Montgomery noticed that the x-coordinate of 2P does not depend on the y-coordinate of P
3. Fast Scalar Multiplication without pre-computation.

Research on Binary ECC Curves

4. Lopez and Dahab Projective Transformation to Reduce Invertions
   – To replace inversions by the multiplication operations and then perform one inversion at the end
5. Mixed Coordinates Addition: Number of multiplications reduced by 10%
6. Parallelization Techniques: for **Point Doubling and** Point Addition
7. Half and Add Technique for Scalar Multiplication

Benefits of ECC

- Same benefits of the other cryptosystems: confidentiality, integrity, authentication and non-repudiation but…
- Shorter key lengths
  - Encryption, Decryption and Signature Verification speed up
  - Storage and bandwidth savings