



IN410
Cryptography and Secure Communications

Ahmad Fadlallah

Course Syllabus



- Context
- History/ Classical ciphers
- Symmetric Encryption
- Cryptographic hash functions
- Key Exchange & Public Key Cryptography
- Authentication

Credits



- The following slides are the adaptation of
  - Lecture slides of Lawrie Brown based on William Stallings book “Cryptography and Network Security: Principles and Practice”
  - Lecture Slides of Dan Boneh – Stanford University
  - Lecture slides of Pawel Wocjan – University of Central California
  - Lecture Slides of Ahmed Serhrouchni – Telecom ParisTech
  - Others (References in the note section)



Public Key Cryptography

Outline



- Introduction
- Essential Number Theory for Public Key Cryptography
- RSA Cryptosystem
- El-Gamal Cryptosystem
- Elliptic Curves Cryptosystem



Introduction

Essential Number Theory for Public Key Cryptography
RSA Cryptosystem
El-Gamal Cryptosystem
Elliptic Curves Cryptosystem

Public-Key Cryptography



- Probably **most significant advance** in the 3000-year history of cryptography
- **Complements rather than replaces private key cryptography**

Why Public-Key Cryptography?



- Developed to address two key issues:
  - Key distribution: how to have secure communications in general without having to trust a KDC with your key
  - Digital signatures: how to verify a message comes intact from the claimed sender

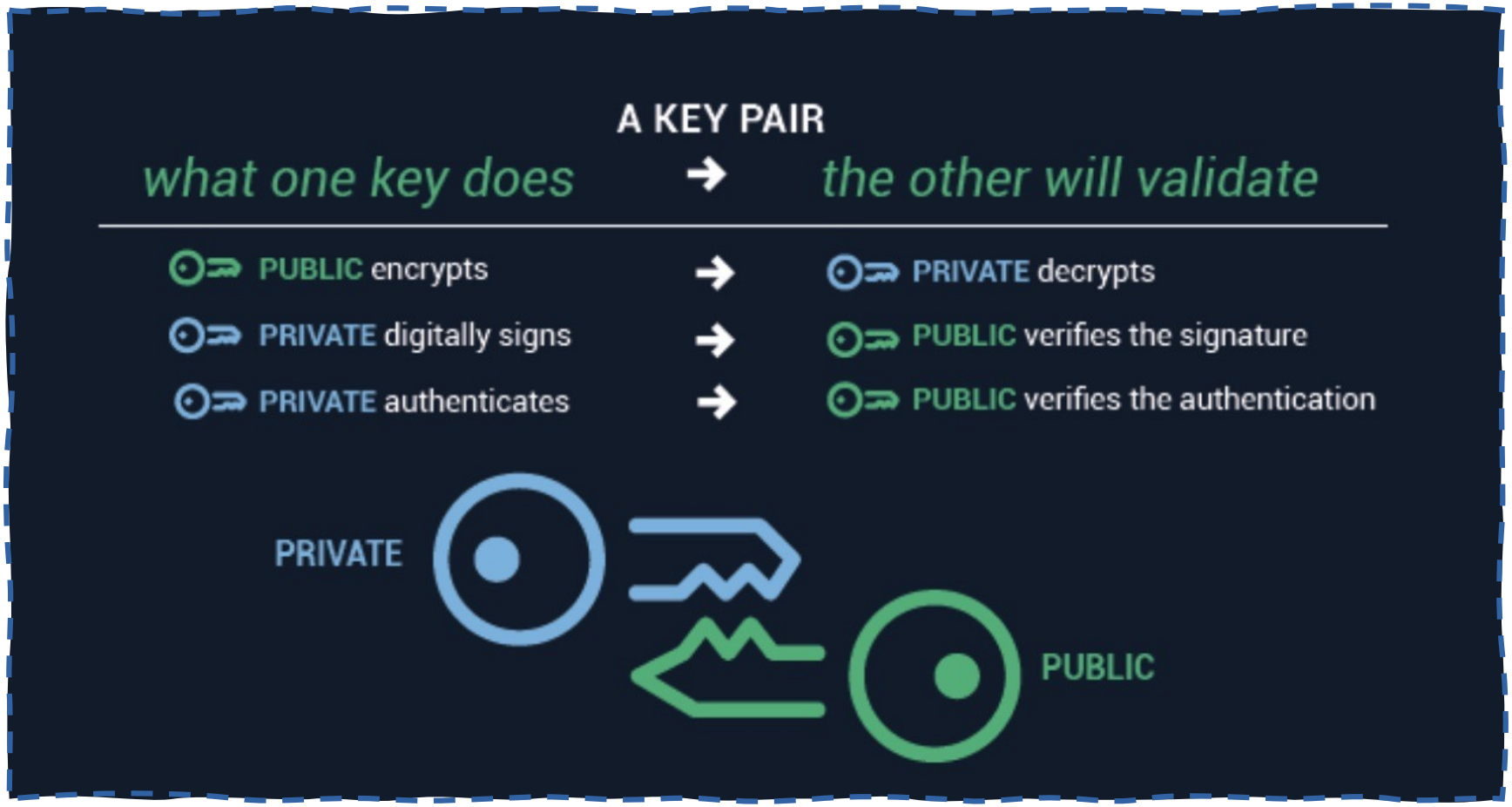
Public-Key Cryptography



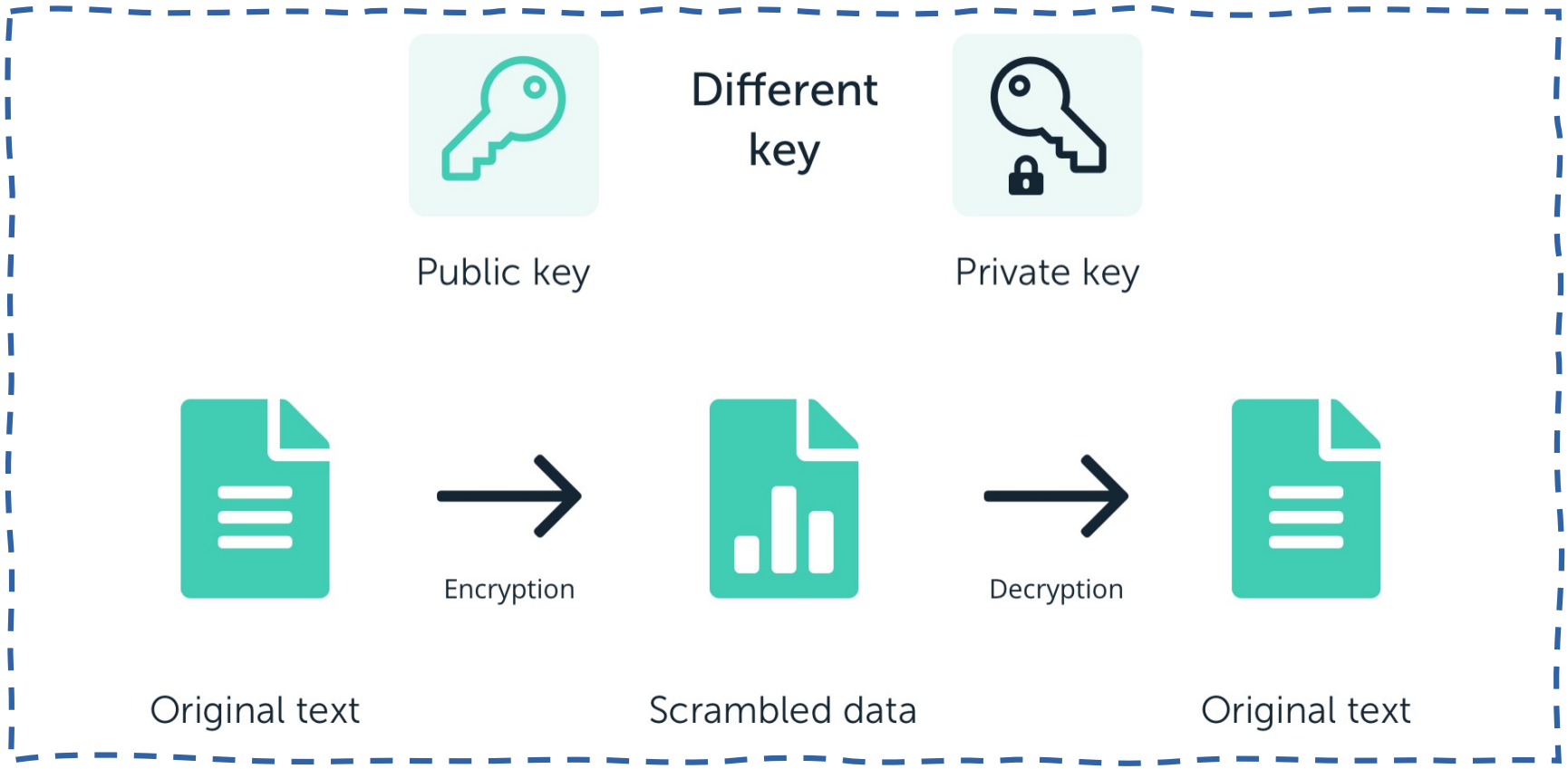
- **Public-key/two-key/asymmetric** cryptography
- Involves the use of two keys:
  - **Public-key**
    - May be known by anybody
    - can be used to encrypt messages, and verify signatures
  - **Private-key**
    - known only to the recipient
    - Used to decrypt messages, and sign (create) signatures
- Infeasible to determine private key from public



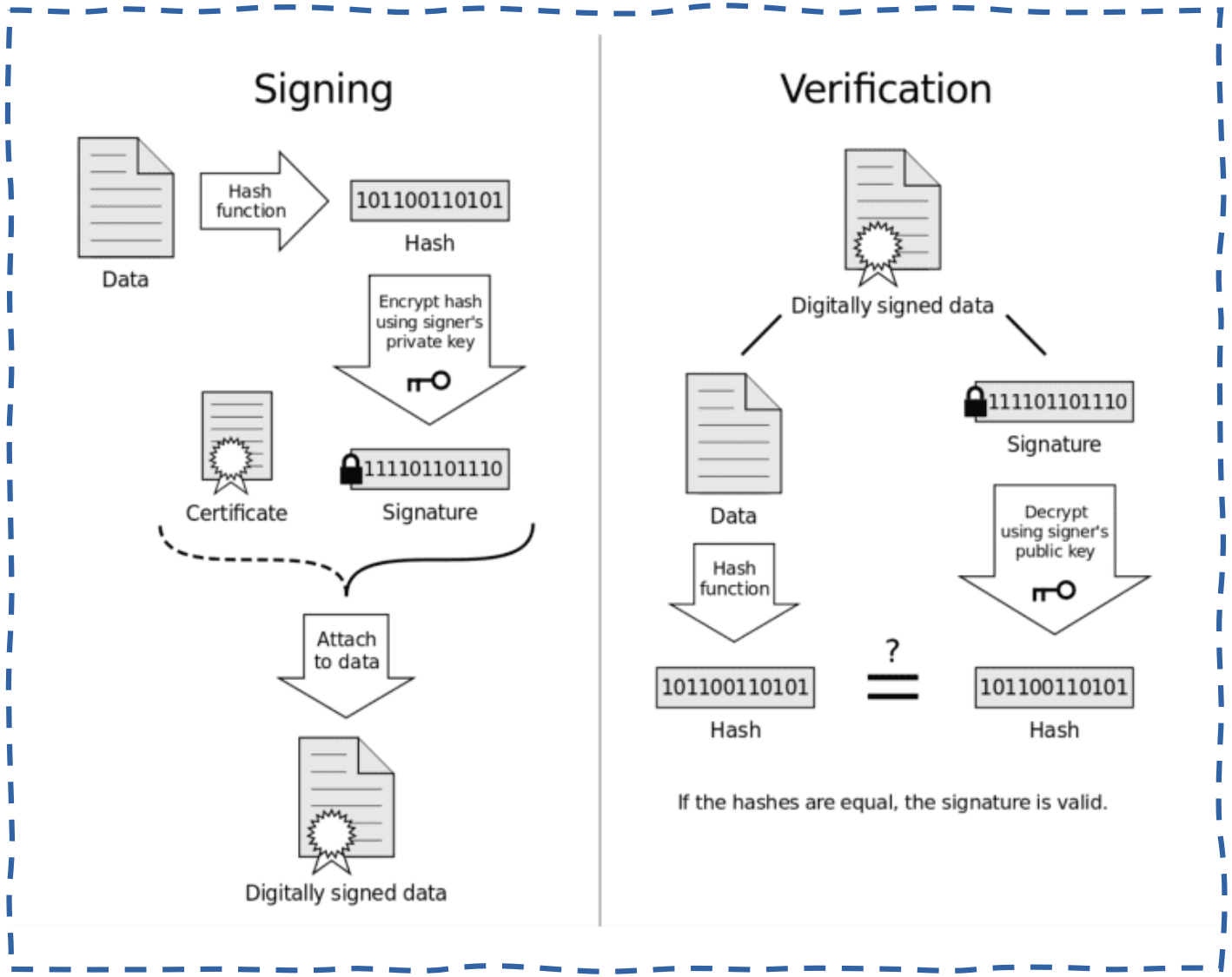
Public-Key Cryptography

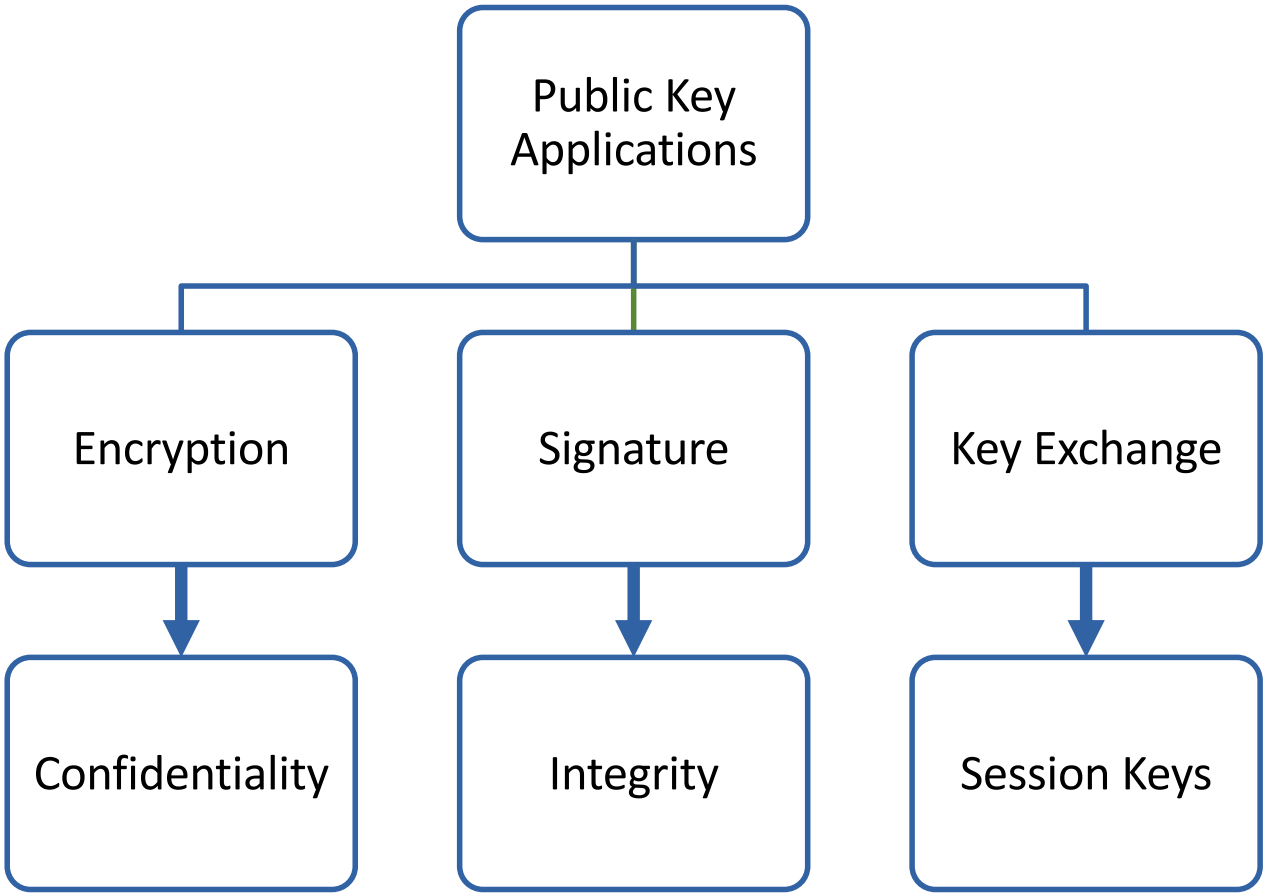


Public-Key Cryptography - Encryption/Decryption

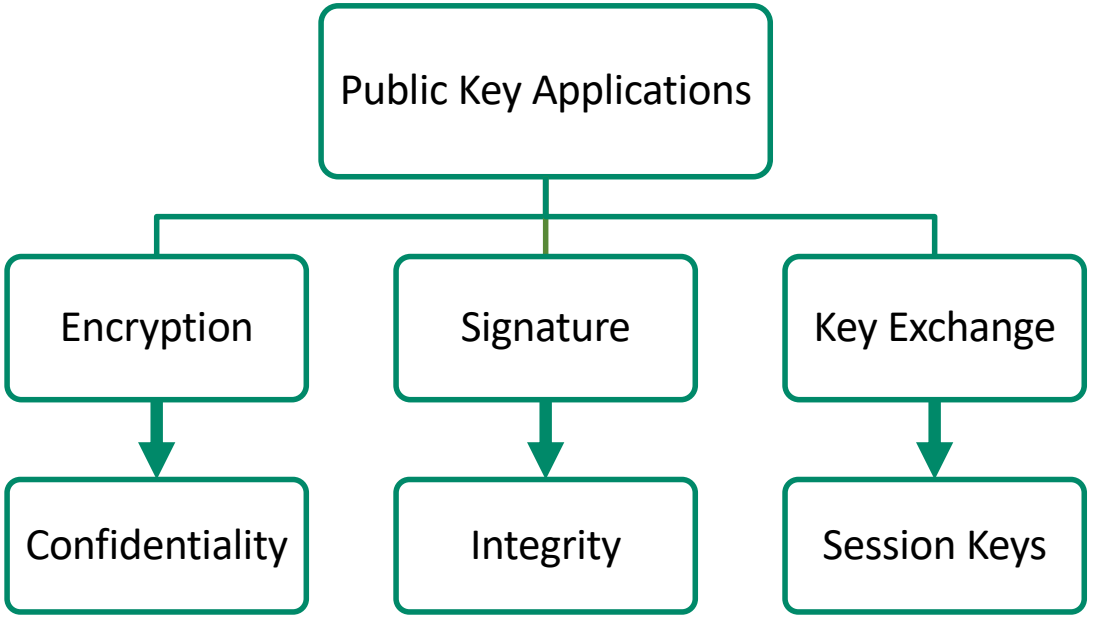


Public-Key Cryptography - Signature/ Verification




Public Key Applications
Encryption
Signature
Key Exchange
Confidentiality
Integrity
Session Keys

Public-Key Applications



Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie–Hellman	No	No	Yes
DSS	No	Yes	No

Public-Key Requirements (Diffie and Hellman)



1. It is computationally easy for B to generate a pair (public key PU_b, private key PR_b).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext: $C = E(PU_b, M)$
3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: $M = D(PR_b, C) = D[PR_b, E(PU_b, M)]$
4. It is computationally infeasible for an adversary, knowing the public key, PU_b, to determine the private key, PR_b
5. It is computationally infeasible for an adversary, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M



Introduction

Essential Number Theory for Public Key Cryptography

RSA Cryptosystem

El-Gamal Cryptosystem

Elliptic Curves Cryptosystem

Euclidean Algorithm



- Compute the Greatest Common Divisor(GCD) of two integers r_0 and r_1: $GCD(r_0, r_1)$
- GCD is easy for small numbers:
  - Factor r_0 and r_1
  - GCD = highest common factor
  - Example:
    - $r_0 = 84 = 2 \times 2 \times 3 \times 7 = 2^2 \times 3 \times 7$
    - $r_1 = 30 = 2 \times 3 \times 5$
    - The GCD is the product of all common prime factors=>

$$2 \times 3 = 6 = GCD\,(30,84)$$

- **But**: Factoring is complicated (and often infeasible) for large numbers

Euclidean Algorithm (cont'd)



- Recall from discrete Mathematics

$$\text{if } n|a \,\&\, n|b \Rightarrow n\,|(a-b)$$

$\Rightarrow GCD(r_1, r_2) = GCD(r_1 - r_2, r_1)$ (supposing $r_1 > r_2$)

$\Rightarrow GCD(r_1, r_2) = GCD(r_1 \; mod \; r_2, r_2)$

- **Core idea:**
  - *Reduce the problem of finding the GCD of two given numbers to that of the GCD of two smaller numbers*
  - **Repeat process recursively**
  - $GCD(r_1, r_2) = GCD(r_2, r_3) = \cdots = GCD(r_i, 0) = r_i$

Euclidean Algorithm - Exercise



- Calculate GCD(973, 301)

Euclidean Algorithm – Exercise - Solution



- $r_1 = 973, r_2 = 301$
- $GCD(r_1, r_2) = GCD(r_2, r_1 mod\ r_2)$
- $GCD(973, 301) = GCD(301, 973 mod\ 301)$
  - $973/301: q = 3, r = 70$
- $GCD(973, 301) = GCD(301, 70)$
  - $301/70: q = 4, r = 21$
- $GCD(973, 301) = GCD(70, 21)$
  - $70/21: q = 3, r = 7$
- $GCD(973, 301) = GCD(21, 7)$
  - $21/7: q = 3, r = 0$
- $GCD(973, 301) = GCD(7, 0) = 7$

Euclidean Algorithm (cont'd)



q	r_1	r_2	r
	973	301	

Euclidean Algorithm (cont'd)



q	r_1	r_2	r
3	973	301	70
4	301	70	21
3	70	21	7
3	21	7	0

Extended Euclidean Algorithm



- Goal: $Given\ r_1\ and\ r_2, find\ r_2^{-1} mod r_1$
- Bezout Theorem: $\exists s, t, such\ that\ s \times r_1 + t \times r_2 = GCD(r_1, r_2)$
- How: $Write\ GCD(r_1, r_2)\ as:\ s \times r_1 + t \times r_2$
- E.E.A. Steps:
  - $r_1 = 1 * r_1 + 0 * r_2$
  - $r_2 = 0 * r_1 + 1 * r_2$
  - $GCD(r_1, r_2): r_1 = q_2 \times r_2 + r_3 \Rightarrow r_3 = s_3 \times r_1 + t_3 \times r_2$
  - $GCD(r_2, r_3): r_2 = q_3 \times r_3 + r_4 \Rightarrow r_4 = s_4 \times r_1 + t_4 \times r_2$
  - ....
  - $GCD(r_{l-2}, r_{l-1}): r_{l-2} = q_{l-1} \times r_{l-1} + r_l \Rightarrow r_l = s_l \times r_1 + t_l \times r_2$
  - $GCD(r_{l-1}, r_l): r_{l-1} = q_l \times r_l + 0 \Rightarrow r_l = GCD(r_1, r_2)$
  - $s = s_l, t = t_l$

Extended Euclidean Algorithm - Example



- **Find the GCD of $r_1 = 973, r_2 = 301$ and write it as $s \times r_1 + t \times r_2$**
  - $GCD(r_1, r_2) = GCD(r_2, r_1 mod\ r_2)$
  - $GCD(973, 301) = GCD(301, 973 mod\ 301)$
    - $973/301: q = 3, r = 70$

    $\Rightarrow 973 = 3 \times 301 + 70 \Rightarrow 70 = 973 - 3 \times 301$
  - $GCD(973, 301) = GCD(301, 70)$
    - $301/70: q = 4, r = 21$

      $\Rightarrow 21 = 301 - 4 \times 70 = 301 - 4 \times (973 - 3 \times 301)$

    $\Rightarrow 21 = -4 \times 973 + 13 \times 301$
  - $GCD(973, 301) = GCD(70, 21)$
    - $70/21: q = 3, r = 7$

    $\Rightarrow 7 = 70 - 3 \times 21 = (973 - 3 \times 301) - 3 \times (-4 \times 973 + 13 \times 301)$

    $\Rightarrow 7 = 13 \times 973 + (-42) \times 301$
  - $GCD(973, 301) = GCD(21, 7) = GCD\ (7,0) = 7$

    $\mathbf{7 = 13 \times 973 + (-42) \times 301}$

Extended Euclidean Algorithm (cont'd)



- Extend the Euclidean algorithm to find modular inverse of $r_2 mod r_1$
- Note that $GCD(r_1, r_2) = 1$ for the inverse to exist
- E.E.A. computes s, t, and the GCD :
  - $GCD(r_1, r_2) = s \times r_1 + t \times r_2$
- Take the relation $mod r_0$

$$GCD(r_1, r_2)\ mod r_1 = s \times r_1\ mod r_1 + t \times r_2\ mod r_1$$

$$1 = 0 + t \times r_2\ mod r_1 \Rightarrow t \times r_2 mod r_1 = 1$$

$$\Rightarrow t\ is\ the\ inverse\ of\ r_2\ mod\ r_1$$

Extended Euclidean Algorithm/ Exercise



- Calculate the modular inverse of $12\ (mod\ 67)$ using Extended Euclidean Algorithm

Extended Euclidean Algorithm/ Exercise

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	67	12							

$$r_1 = 1 * r_1 + 0 * r_2$$
$$r_2 = 0 * r_1 + 1 * r_2$$

$$s = s_1 - s_2 * q$$
$$t = t_1 - t_2 * q$$

Extended Euclidean Algorithm/ Exercise



q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	67	12	7	1	0	1	0	1	-5
1	12	7	5	0	1	-1	1	-5	6
1	7	5	2	1	-1	2	-5	6	-11
2	5	2	1	-1	2	-5	6	-11	28

Stop Condition

Inverse of $r_1 mod\ r_2$

Inverse of $r_2 mod\ r_1$

$$r_1 = 1 * r_1 + 0 * r_2$$
$$r_2 = 0 * r_1 + 1 * r_2$$

$$s = s_1 - s_2 * q$$
$$t = t_1 - t_2 * q$$



Introduction

Essential Number Theory for Public Key Cryptography

RSA Cryptosystem

El-Gamal Cryptosystem

Elliptic Curves Cryptosystem

RSA



- Proposed by Rivest, Shamir & Adleman of MIT in 1977
- Best known & widely used public-key scheme
- Based on exponentiation in a finite (galois) field over integers modulo a prime
- Uses large integers (e.g. 2048 bits)
- Security due to cost of factoring large numbers (Hard problem)

RSA En/decryption



- To encrypt a message M the sender:
  - obtains public key of recipient $PU = \{e, n\}$
  - computes: $C = M^e \ mod \ n$, where $0 \leq M < n$
- To decrypt the ciphertext C the owner:
  - Uses their private key $PR = \{d, n\}$
  - Computes: $M = C^d \ mod \ n$
- Note: message M must be smaller than the modulus n (block if needed)

RSA Key Setup



- Each user generates a public/private key pair by:
  - Selecting two large primes at random: p, q
  - Computing their system modulus $n = p \times q$
    - note $\phi(n) = (p-1)(q-1)$
  - Selecting at random the encryption key e
    - where $1 < e < \phi(n), \quad GCD(e, \phi(n)) = 1$
  - Solve following equation to find decryption key d
    - $e.d = 1 \; mod \; \phi(n)$ and $0 \leq d \leq n$
- Publish their public encryption key: $PU = \{e, n\}$
- Keep secret private decryption key: $PR = \{d, n\}$

Why RSA Works



Euler's Theorem:

$$a^{\phi(n)} mod\ n\ =\ 1\ ,\ \text{where}\ GCD(a,n) = 1$$

e & d are inverses $mod\ \emptyset(n)$

$$\Rightarrow e.d = 1 + k.\emptyset(n)\ \text{for some}\ \mathbf{k}$$

$$C^d\ =\ M^{ed}\ =\ M^{1+k\phi(n)}\ = M * (M^{\phi(n)})^k$$

$$=\ M^1.(1)^k\ \overset{=}{}\ M^1\ =\ M\ mod\ n$$

RSA Example - Key Setup



1. Select primes: $p = 17$ & $q = 11$
2. Calculate: $n = p \times q = 17 \times 11 = 187$
3. Calculate: $\varphi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e: $GCD(e, 160) = 1$; choose $e = 7$
5. Determine d: $de = 1 \bmod 160$ and $d < 160$.

   $d = 23$ since $23 \times 7 = 161 = 1 \times 160 + 1$

1. Publish public key $PU = \{7,187\}$
2. Keep secret private key $PR = \{23,187\}$

RSA Example - En/Decryption



- Sample RSA encryption/decryption is:
  - Given message $M = 88$ (NB $88 < 187$)
  - Encryption:

$$C = 88^7 \; mod \; 187 = 11$$

  - Decryption:

$$M = 11^{23} \; mod \; 187 = 88$$

RSA Exercise



- Alice wants to generate an RSA key pair using the following parameters: p=3, q=11 and e=7.
  - What is the public key of Alice?
  - Calculate the private key of Alice. Detail your answer
  - Bob wants to send the message m=2 encrypted to Alice. Explain the encryption process.
  - Explain how Alice decrypts the ciphertext she receives from Bob.

Exercise



- A professor sends the student grades to the school secretary by email. The grades are encrypted using RSA asymmetric cryptography. The public key of the professor is (3,55), that of the secretary is (3,33).
  - Find the private key of the professor and the secretary.
  - To ensure the confidentiality of the message, the professor encrypts the grades with the secretary RSA public key. What is the encrypted message associated with the grade 12?
  - To ensure the authenticity of the messages, the professor signs each grade with his private key and encrypts the result with the secretary public key. The secretary receives the message 23. What is the corresponding grade?



Introduction

Essential Number Theory for Public Key Cryptography

RSA Cryptosystem

El-Gamal Cryptosystem

Elliptic Curves Cryptosystem

El-Gamal Cryptography



- Public-key cryptosystem related to D-H
- Proposed by Taher el-Gamal
- Uses exponentiation in a finite (Galois) Field
- Security based on difficulty of computing discrete logarithms, as in D-H
- Key Generation: Each user (e.g. Alice)
  - Chooses a **secret key** (number): $1 < a < p - 1$
  - compute their **public key**: $A = g^a \, mod \, p$
  - Share their public key with other users

El-Gamal Message Encryption



- Bob encrypts a message to send to Alice
  - Represent message M in range $0 \leq M \leq p-1$
    - Longer messages must be sent as blocks
  - Choose random integer r with $1 \leq r \leq p-1$
  - Compute one-time key $K = A^r \, mod \, p$
  - Encrypt M as a pair of integers (C_1, C_2)
    - $C_1 = g^r \, mod \, p \; ; C_2 = K.M \, mod \, p$

El-Gamal Message Decryption



- Alice then recovers message by:
  - Recovering key K as: $K = C_1^a \ mod \ p$
  - Calculate $K^{-1} \ mod \ p$
  - Computing M as $M \ = \ C_2 \ K^{-1} \ mod \ p$
- A unique r must be used each time otherwise the result is insecure

El-Gamal Example



- Bob wants to send a message $m = 17$ to Alice using El-Gamal Encryption Scheme
- Alice uses field: $p = 19$ and $g = 10$ and chooses $a = 5$ as her secret key
- Bob chooses $r = 6$ as a random key
- Detail the encryption and decryption scenario

El-Gamal Example - Solution



- $m = 17, p = 19, g = 10, a = 5, r = 6$
- Alice computes her public key:
  - $A = g^a \; mod \; p \; = \; 105 \; mod \; 19 \; = \; 3$
- Bob sends message $m = 17$ as $(11,5)$ by computing:
  - $K \; = \; A^r \; mod \; p \; = \; 36 \; mod \; 19 \; = \; 7$
  - $C_1 \; = \; g^r \; mod \; p \; = \; 106 \; mod \; 19 \; = \; 11$
  - $C_2 = K \times M \; mod \; p \; = \; 7 \times 17 \; mod \; 19 \; = \; 5$

El-Gamal Example – Solution (cont.)



- Alice recovers original message by computing:
  - $K = C_1^a \, mod \, p = 115 \, mod \, 19 = 7$
  - $K^{-1} = 7^{-1} = 11$
  - $M = C_2 \, K^{-1} \, mod \, p = 5 \times 11 \, mod \, 19 = 17$

El-Gamal Encryption - Exercise



- Let $p = 53, g = 2, B = 30$ the public key El-Gamal of Bob.
- Encrypt the message $M = 42$ with Bob's public key (consider a random secret key of 3)
- Assume that Bob's secret key is 13.
  - Decrypt the message $(C_1 = 15, C_2 = 17)$



Introduction
Essential Number Theory for Public Key Cryptography
RSA Cryptosystem
El-Gamal Cryptosystem

Elliptic Curves Cryptosystem

Elliptic Curve Cryptography



- Majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials
- Imposes a significant load in storing and processing keys and messages
- An alternative is to use elliptic curves
- Offers same security with smaller bit sizes

Comparable Key Sizes for Equivalent Security



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

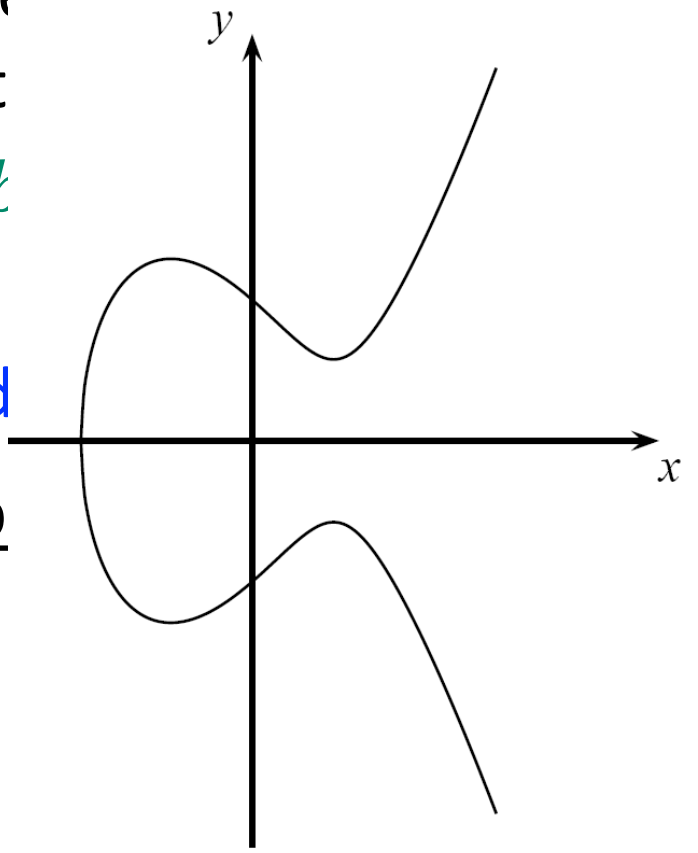
Computations on Elliptic Curves



- Elliptic curves are polynomials that define points based on the (simplified) *Weierstrass* equat

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

- Parameters a, b specify the shape of
- Elliptic curves can not just be defined numbers $\mathbb{R}$ but over many other typ



$Example: y^2 = x^3 - 3x + 3$ *over* $\mathbb{R}$

Computations on Elliptic Curves (cont'd)



In cryptography, we are interested in elliptic curves modulo a prime p

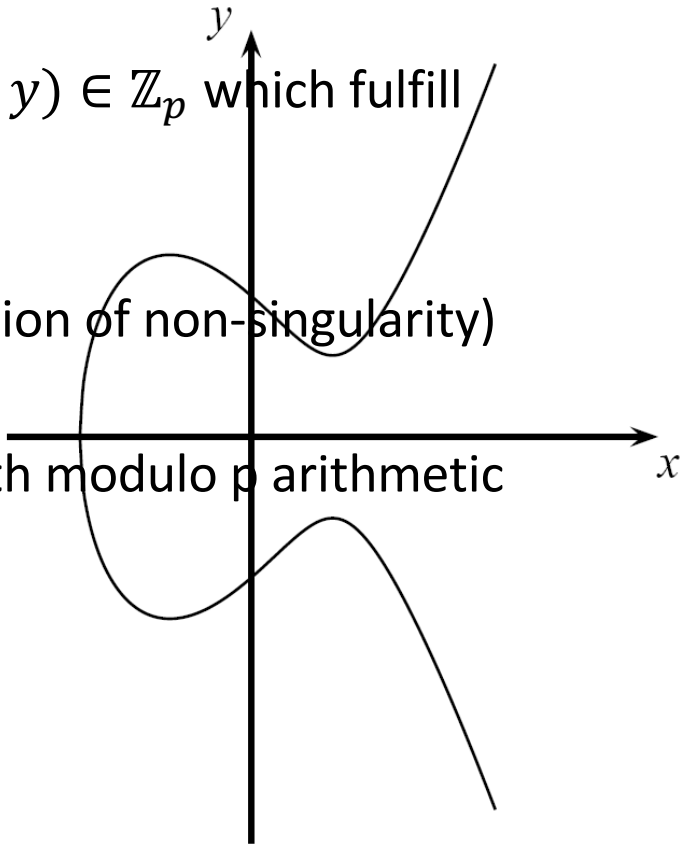
Definition: Elliptic Curves over prime fields

The elliptic curve over $\mathbb{Z}_p$, p>3 is the set of all pairs $(x, y) \in \mathbb{Z}_p$ which fulfill

$$y^2 = x^3 + ax + b \ (mod\ p)$$

together with an imaginary point of infinity θ,

where $a, b \in \mathbb{Z}_p$ and $4a^3 + 27b^2 \neq 0 \ (modp)$ (condition of non-singularity)



Note that $\mathbb{Z}_p = \{0,1, \ldots., p-1\}$ is a set of integers with modulo p arithmetic

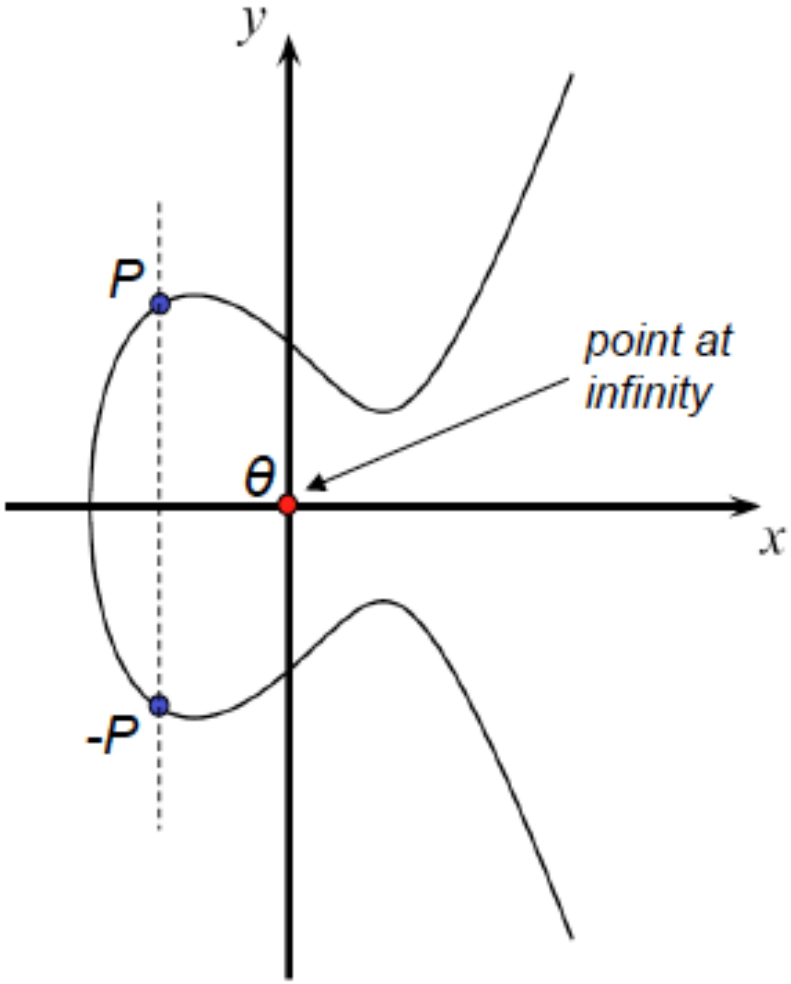
Computations on Elliptic Curves (cont'd)



- Identity Point
  - denoted by θ
  - a special element required to all operation,

$$\forall P \in E: P + \theta = P = \theta$$

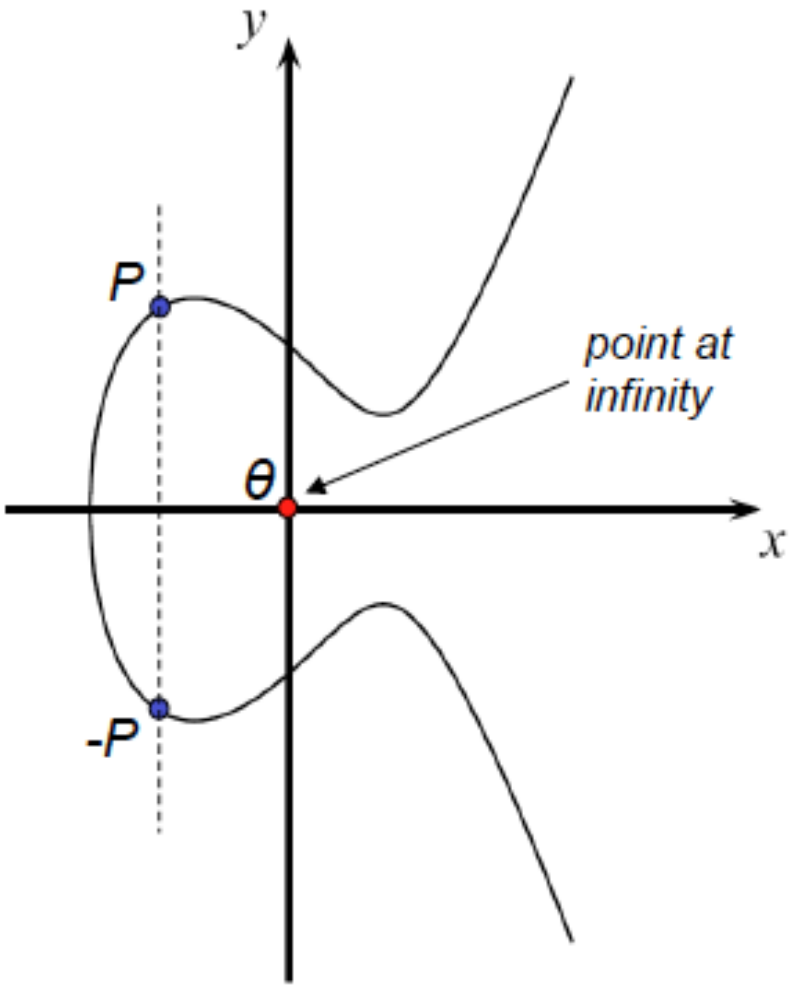
  - Not on the curve
  - Additionally added to the group



Computations on Elliptic Curves (cont'd)



- Elliptic Curves are symmetric along t
- Up to two solutions y and -y exist fo curve
- For each point $P = (x, y)$, the inve defined as $-P = (x, -y)$



Computations on Elliptic Curves (cont'd)

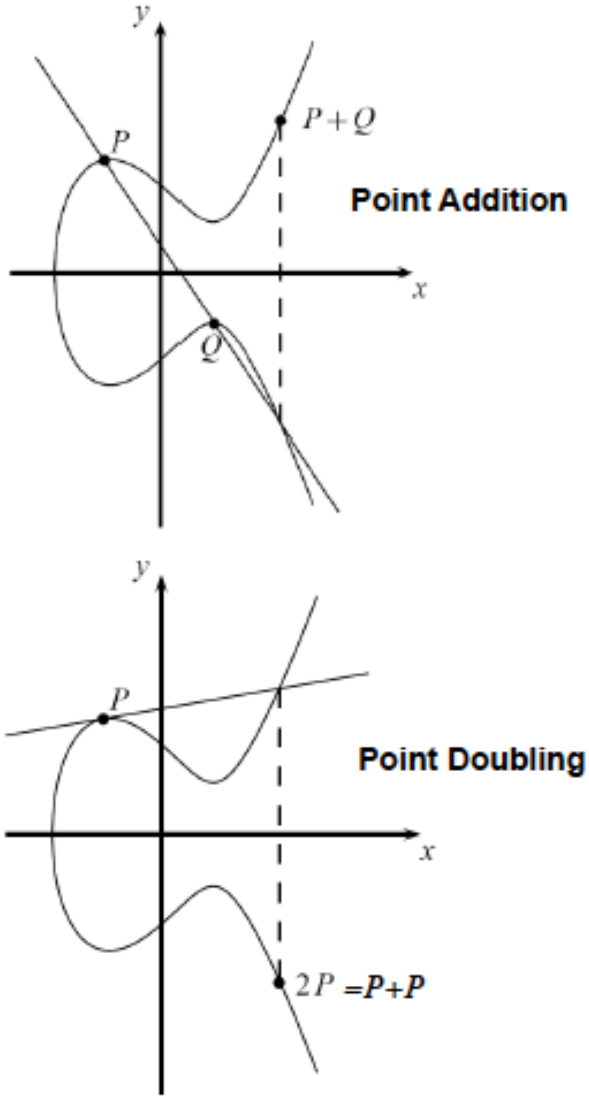


- Generating a group of points on elliptic cur
  addition operation

$$P + Q = R$$

$$(x_P, y_P) + (x_Q, y_Q) = (x_R,$$

- Geometric Interpretation of point addition
  - Draw straight line through P and Q
    - if P=Q use tangent line instead
  - Mirror third intersection point of drawn lin curve along the x-axis



Computations on Elliptic Curves (cont'd)

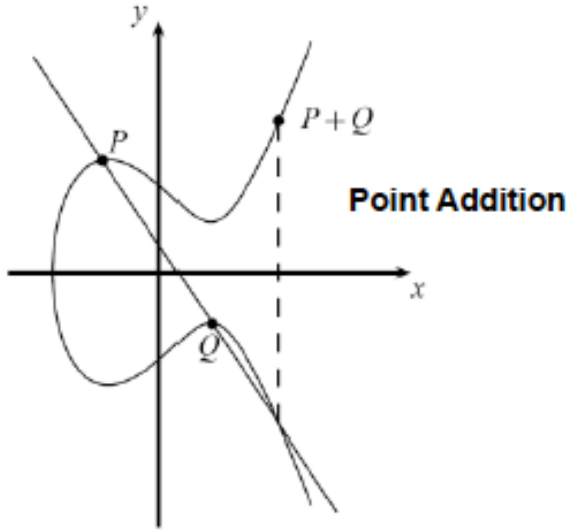


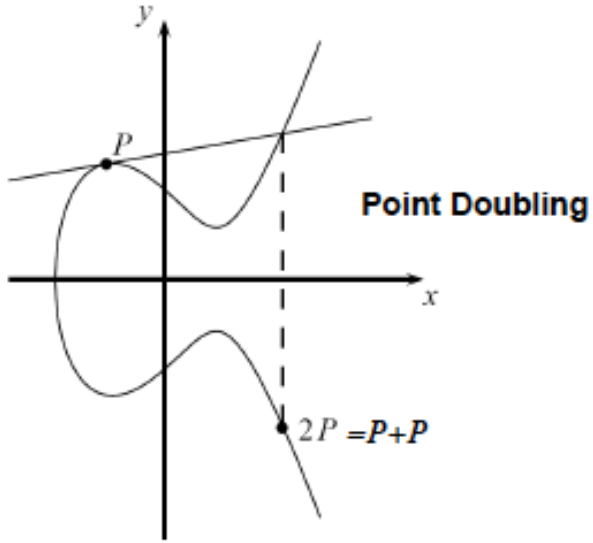
- Elliptic Curve Point Addition and Doubl

$x_3 = s^2 - x_1 - x_2 \bmod p$ and $y_3 = s(x_1 - x_3) - y_1 \bmod p$

where

$$s = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p & \text{; if P} \neq \text{Q (point addition)} \\ \dfrac{3x_1^2 + a}{2y_1} \bmod p & \text{; if P = Q (point doubling)} \end{cases}$$





Computations on Elliptic Curves (cont'd)



- Exercise:
  - Given E: $y^2 = x^3 + 2x + 2 \; mod 17$ and point $P = (5,1)$
  - Goal: Compute $2P = P + P = (5,1) + (5,1) = (x_3, y_3)$

$x_3 = s^2 - x_1 - x_2 \bmod p$ and $y_3 = s(x_1 - x_3) - y_1 \bmod p$

where

$$s = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \bmod p & \text{; if P} \neq \text{Q (point addition)} \\ \dfrac{3x_1^2 + a}{2y_1} \bmod p & \text{; if P = Q (point doubling)} \end{cases}$$

Computations on Elliptic Curves (cont'd)



- Exercise:
  - Given E: $y^2 = x^3 + 2x + 2 \; mod 17$ and point $P = (5,1)$
  - Goal: Compute 2P = P+P = (5,1)+(5,1)= (x_3,y_3)

$$s = \frac{3x_1^2 + a}{2y_1} = (2 \cdot 1)^{-1}(3 \cdot 5^2 + 2) = 2^{-1} \cdot 9 \equiv 9 \cdot 9 \equiv 13 \bmod 17$$

$$x_3 = s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 159 \equiv 6 \bmod 17$$

$$y_3 = s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 \equiv 3 \bmod 17$$

$$2P = (5,1) + (5,1) = (6,3)$$

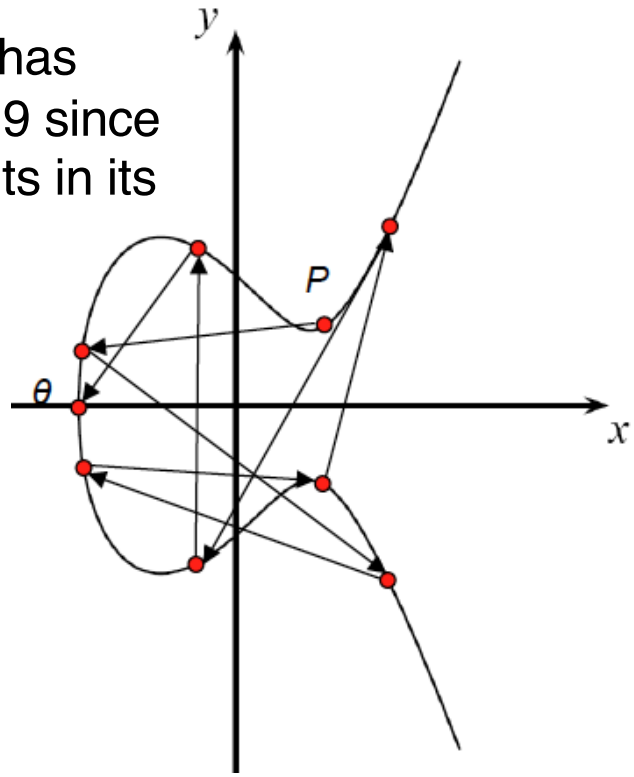
Computations on Elliptic Curves (cont'd)



- The points on an elliptic curve and the point at infinity θ form **cyclic subgroups**

$2P = (5,1)+(5,1) = (6,3)$	$11P = (13,10)$
$3P = 2P+P = (10,6)$	$12P = (0,11)$
$4P = (3,1)$	$13P = (16,4)$
$5P = (9,16)$	$14P = (9,1)$
$6P = (16,13)$	$15P = (3,16)$
$7P = (0,6)$	$16P = (10,11)$
$8P = (13,7)$	$17P = (6,14)$
$9P = (7,6)$	$18P = (5,16)$
$10P = (7,11)$	$19P = \theta$

This elliptic curve has **order** #E = |E| = 19 since it contains 19 points in its cyclic group.



Number of Points on an Elliptic Curve



- How many points can be on an arbitrary elliptic curve?
  - Example: $E: y^2 = x^3 + 2x + 2 \; mod \; 17$ has 19 points
  - However, determining the point count on elliptic curves in general is hard
- Hasse's theorem bounds the number of points to a restricted interval
  - Given an elliptic curve module p, the number of points on the curve is denoted by $\#E$ and is bounded by

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}$$

  - Interpretation: The number of points is "close to" the prime p
  - Example: To generate a curve with about 2^{160} points, a prime with a length of about 160 bits is required

Elliptic Curve Discrete Logarithm Problem



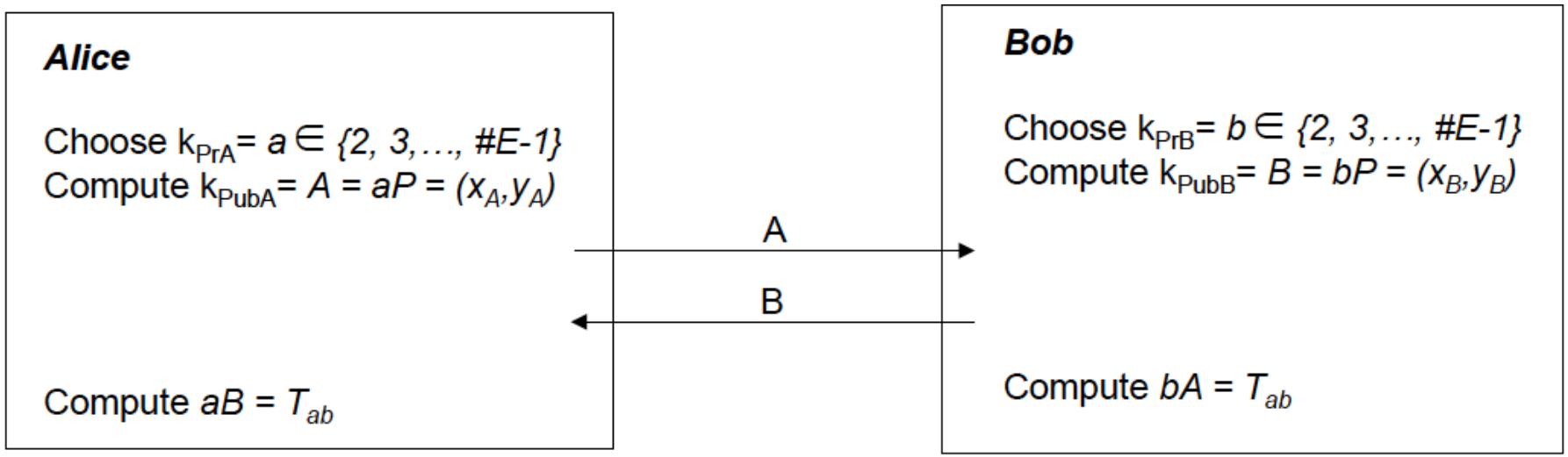
- Cryptosystems rely on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP)
- Definition:
  - Given a primitive element P and another element T on an elliptic curve E.
  - The ECDL problem is finding the integer d, where $1 \leq d \leq \#E$ such that: $\underbrace{P + P + \cdots + P}_{d\ times} = dP = T$

Cryptosystems are based on the idea that d is large and kept secret and attackers cannot compute it easily

Elliptic Curve Diffie-Hellman Key Exchange (ECDH)



- Given a prime p, a suitable elliptic curve E and a point $P = (x_P, y_P)$



- **Joint secret between Alice and Bob**: $T_{AB} = (x_{AB}, y_{AB})$
- **Proof for correctness:**
  - Alice computes $aB = a(bP) = abP$
  - Bob computes $bA = b(aP) = abP$ since group is associative

Elliptic Curve Diffie-Hellman Key Exchange (ECDH)



- The ECDH is often used to derive session keys for (symmetric) encryption
- One of the coordinates of the point T_{AB} (usually the x-coordinate) is taken as session key

