Information Security

CS3002

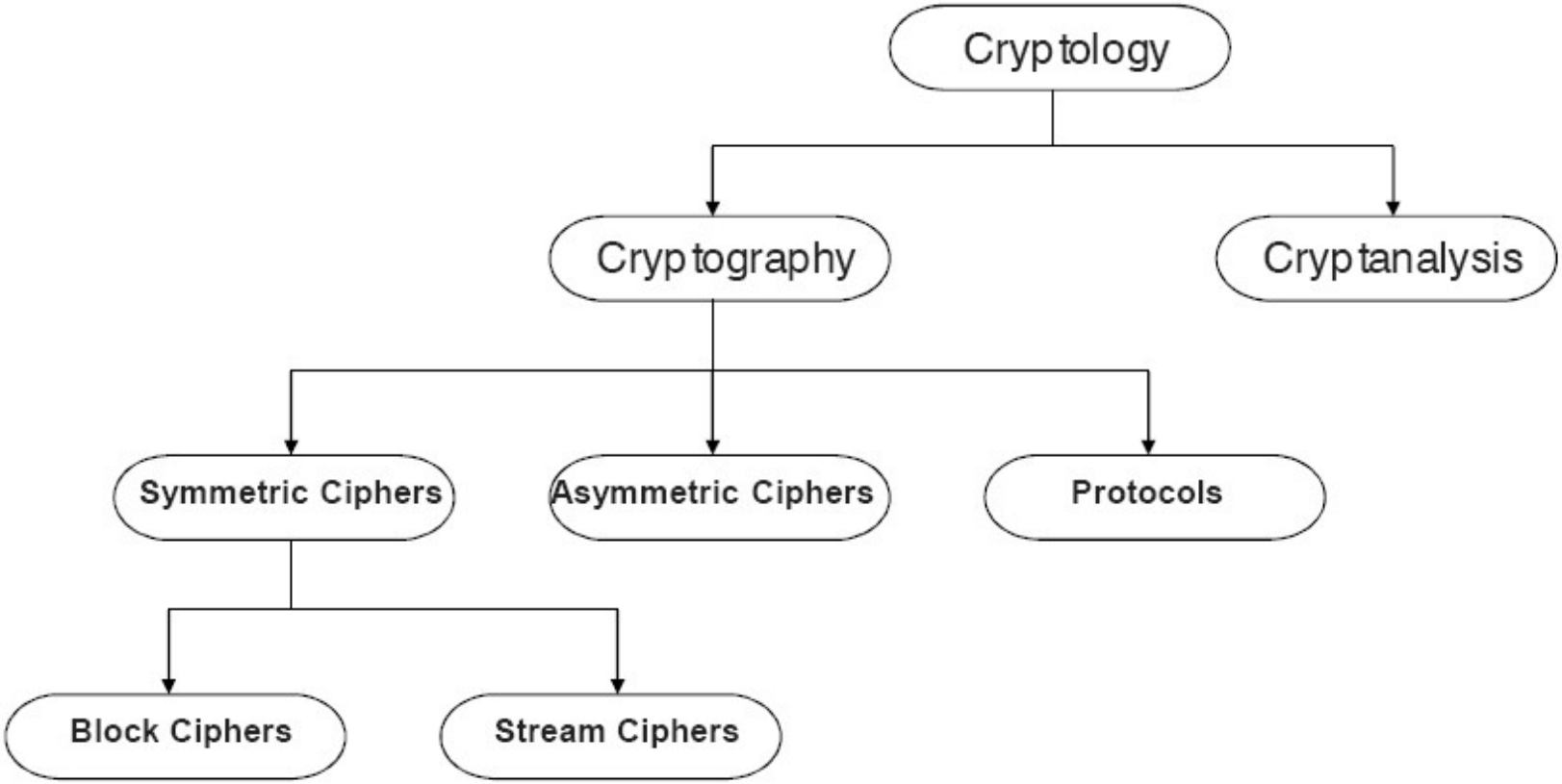
Lecture 6

11th September 2023

Dr. Rana Asif Rehman

Email: r.asif@lhr.nu.edu.pk

Classification of the Field of Cryptology



Adopted with thanks from: Chapter 1 of Understanding Cryptography by Christof Paar and Jan Pelzl

Private-Key Cryptography

- Traditional **private/secret/single key** cryptography uses **one** key.
- Shared by both sender and receiver.
- If this key is disclosed communications are compromised.
- Also is **symmetric**, parties are equal.
  - Hence does not protect sender from receiver forging a message & claiming is sent by sender.

Public-Key Cryptography

- Probably most significant advance in the 3000 year history of cryptography.
- Uses **two** keys – a public & a private key.
- **Asymmetric** since parties are **not** equal.
- Uses clever application of number theoretic concepts to function.
- Complements **rather than** replaces private key cryptography.

Public-Key Cryptography

- **Public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - A **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures.**
  - A **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures.**
- Is **asymmetric** because
  - Those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures.

Need for Both

There is a very important fact that is sometimes misunderstood: The advent of asymmetric-key cryptography does not eliminate the need for symmetric-key cryptography.

Public-Key Cryptography

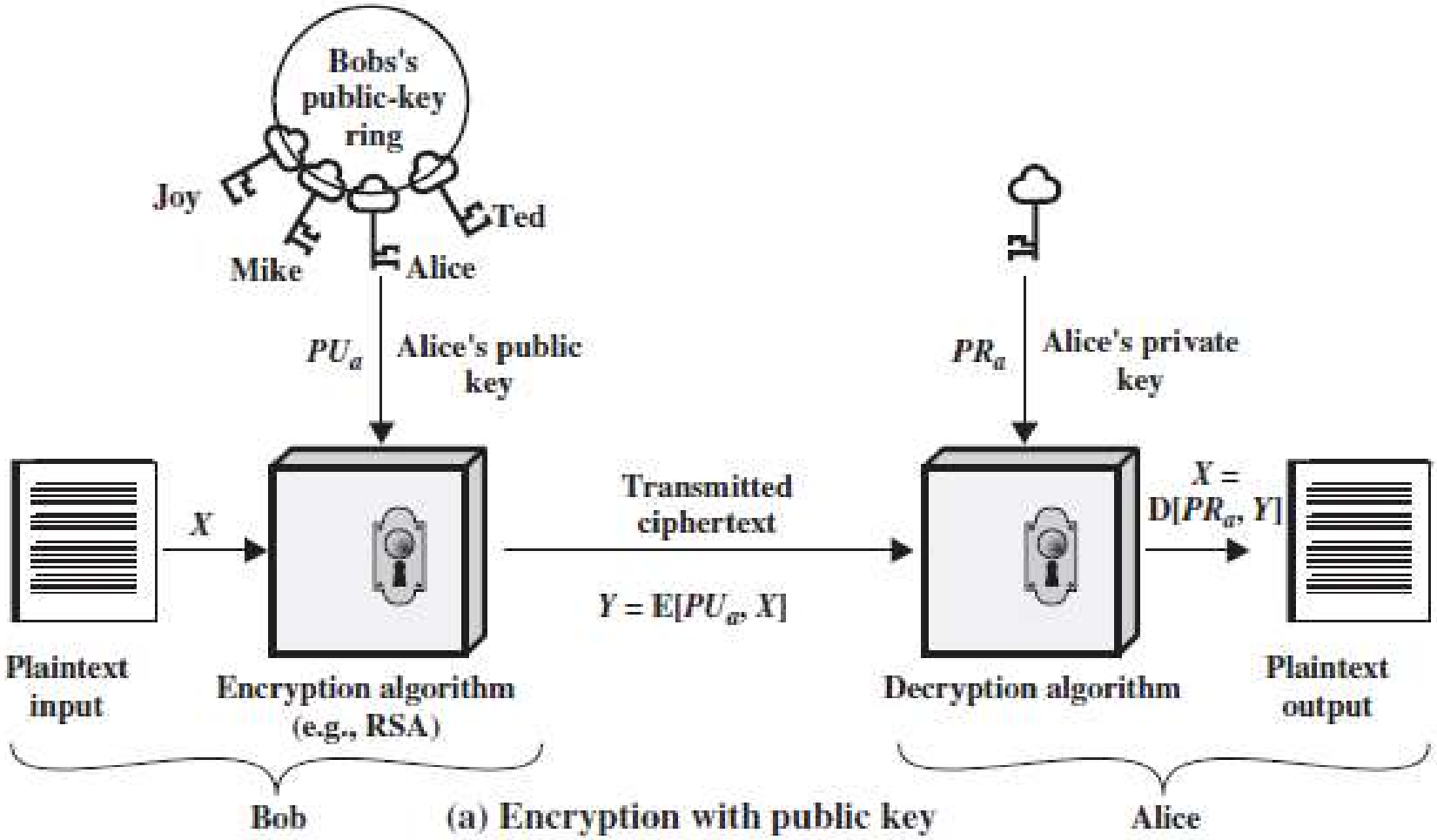
Plaintext/Ciphertext

Unlike in symmetric-key cryptography, plaintext and ciphertext are treated as integers in asymmetric-key cryptography.

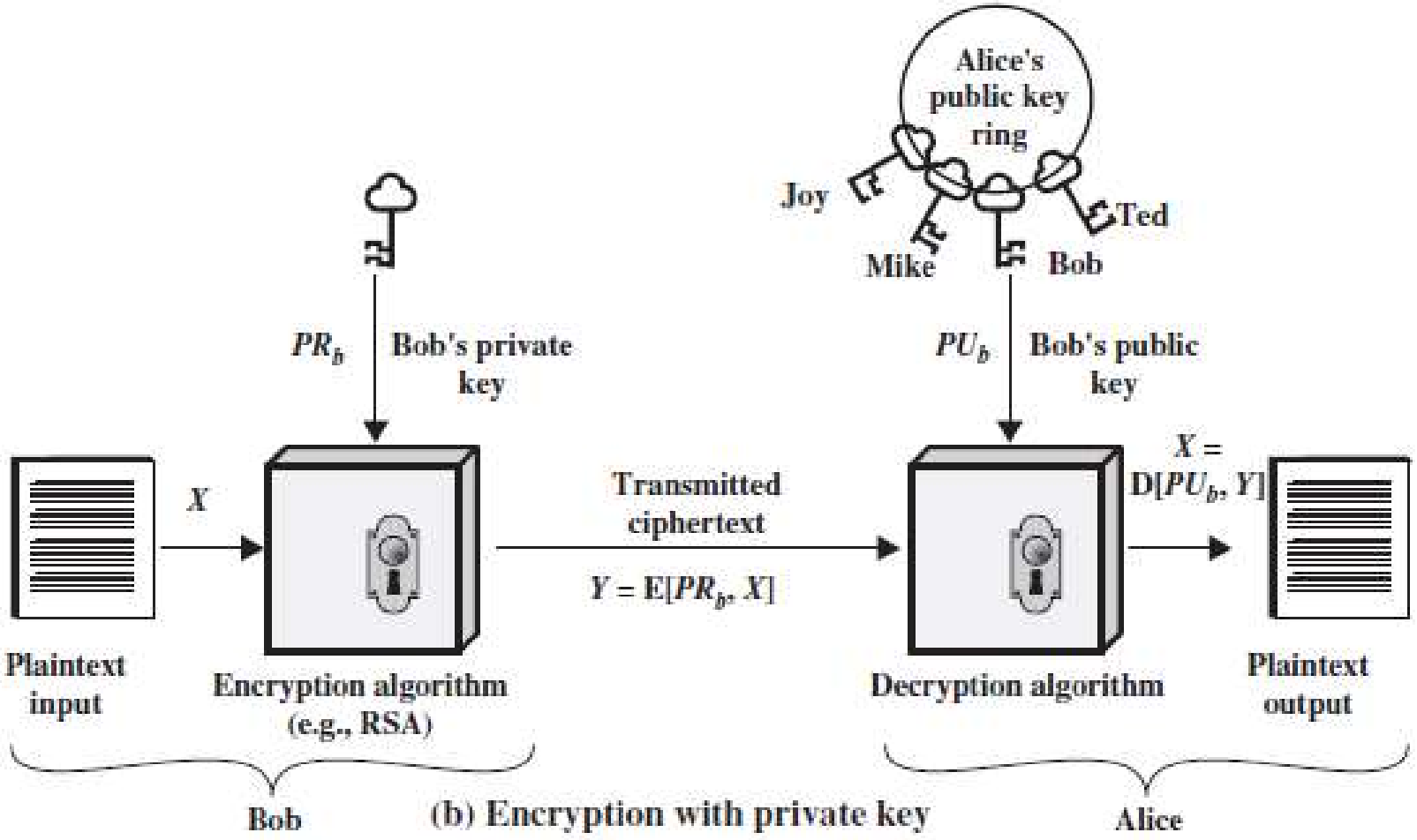
Encryption/Decryption

$$C = f(K_{public}, P) \qquad P = g(K_{private}, C)$$

Encryption


Bobs's public-key ring
Joy
Ted
Mike
Alice
PU_a
Alice's public key
PR_a
Alice's private key
X
Transmitted ciphertext
$Y = E[PU_a, X]$
$X = D[PR_a, Y]$
Plaintext input
Encryption algorithm (e.g., RSA)
Decryption algorithm
Plaintext output
Bob
(a) Encryption with public key
Alice

Authentication



Public-Key Characteristics

- Public-Key algorithms rely on two keys where:
  - It is computationally infeasible to find decryption key knowing only algorithm & encryption key.
  - It is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known.
  - Either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms).

Operation

- Each user generates pair of keys.
- Place one of keys in public register or accessible file (public key).
- Keep other companion key (public key).
- If Bob wants to send confidential message to Alice: encrypt with Alice's public key.
- Only Alice can decrypt message with her private key.

Advantages

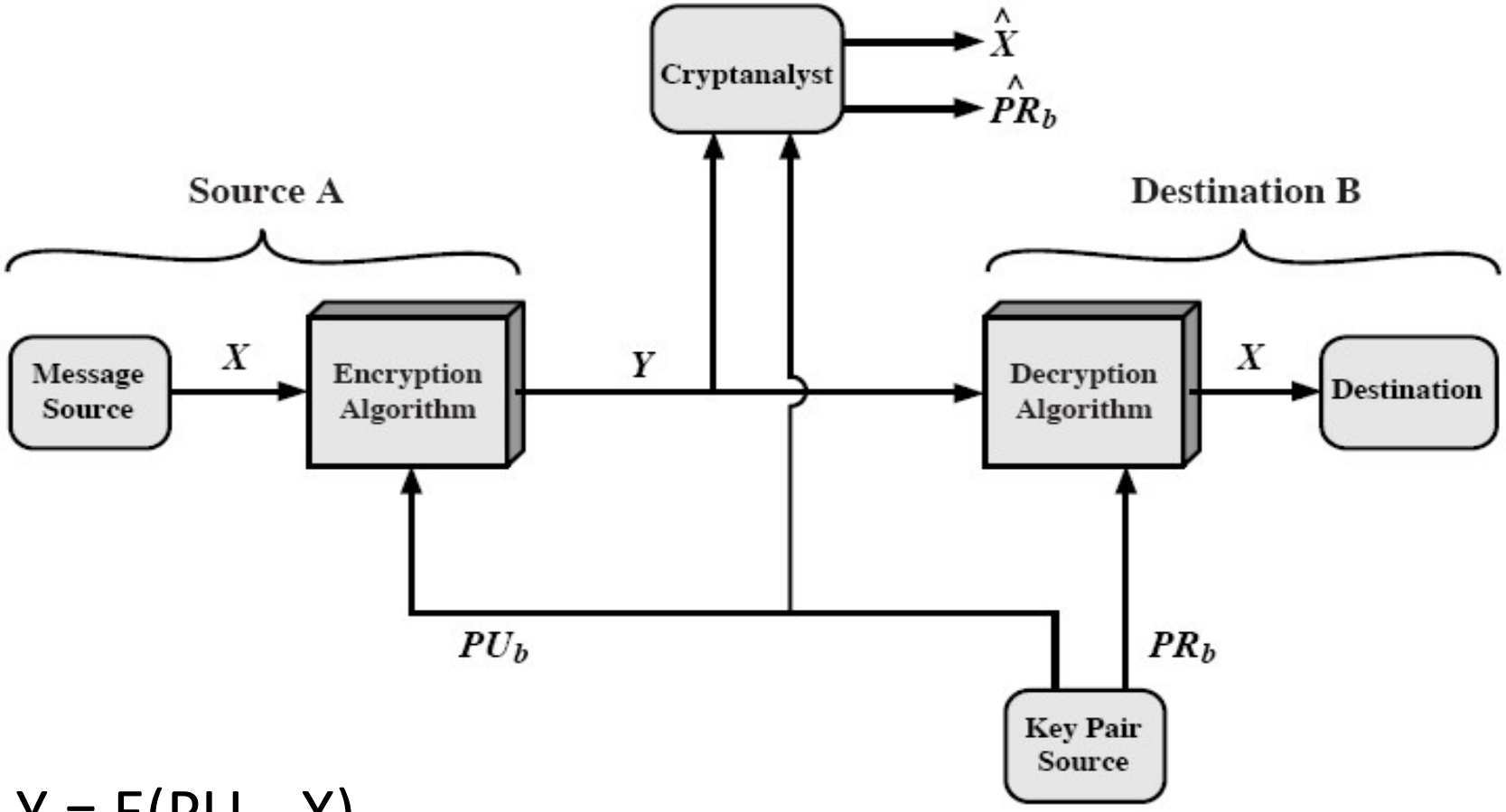
- Private keys generated locally.
- Private key need not to be distributed.
- Keys can be changed at any time.
  - At any time, a system can change its private key and publish the companion public key to replace its old public key.

Public-Key Applications

- Can classify uses into 3 categories:
  - **Encryption/decryption** (provide secrecy)
  - **Digital signatures** (provide authentication)
  - **Key exchange** (of session keys)
- Some algorithms are suitable for all uses, others are specific to one.

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

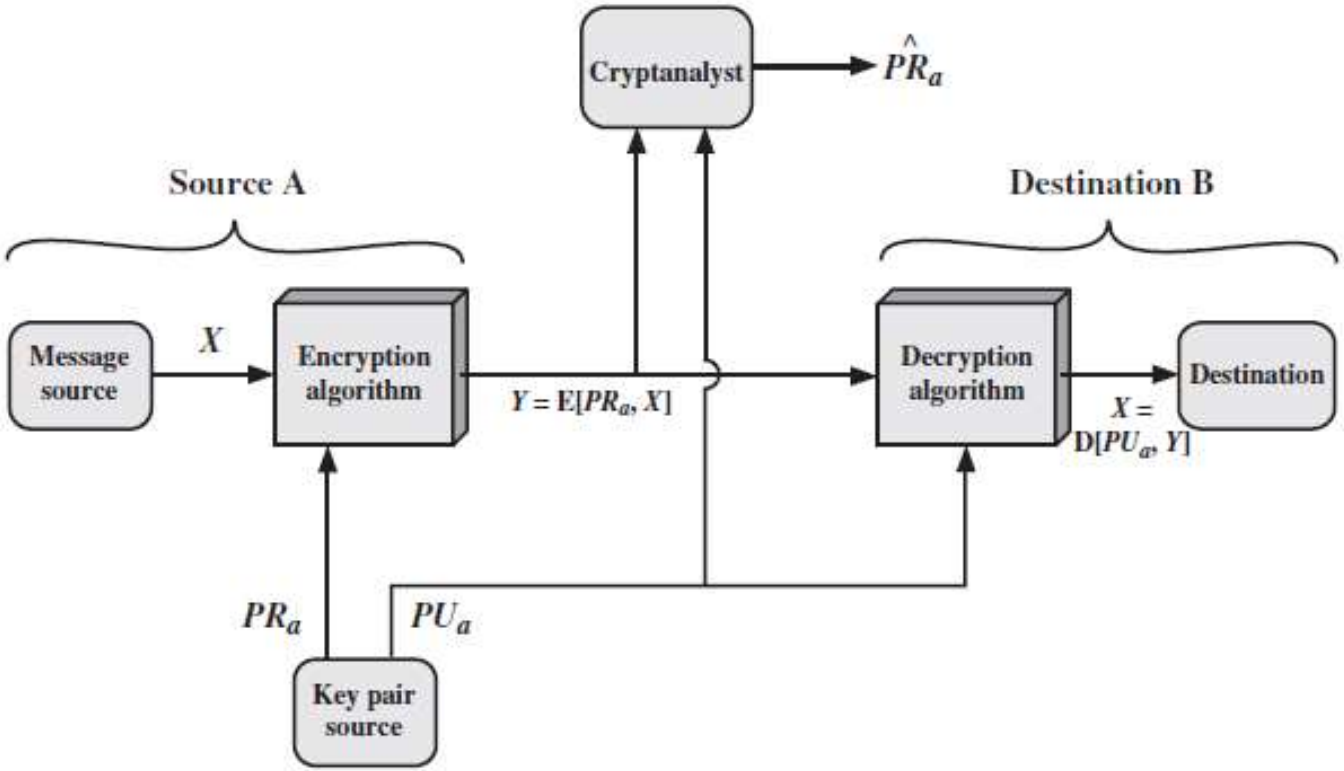
Applications: Confidentiality



$Y = E(PU_b, X)$

$X = D(PR_b, Y)$

Applications: Authentication



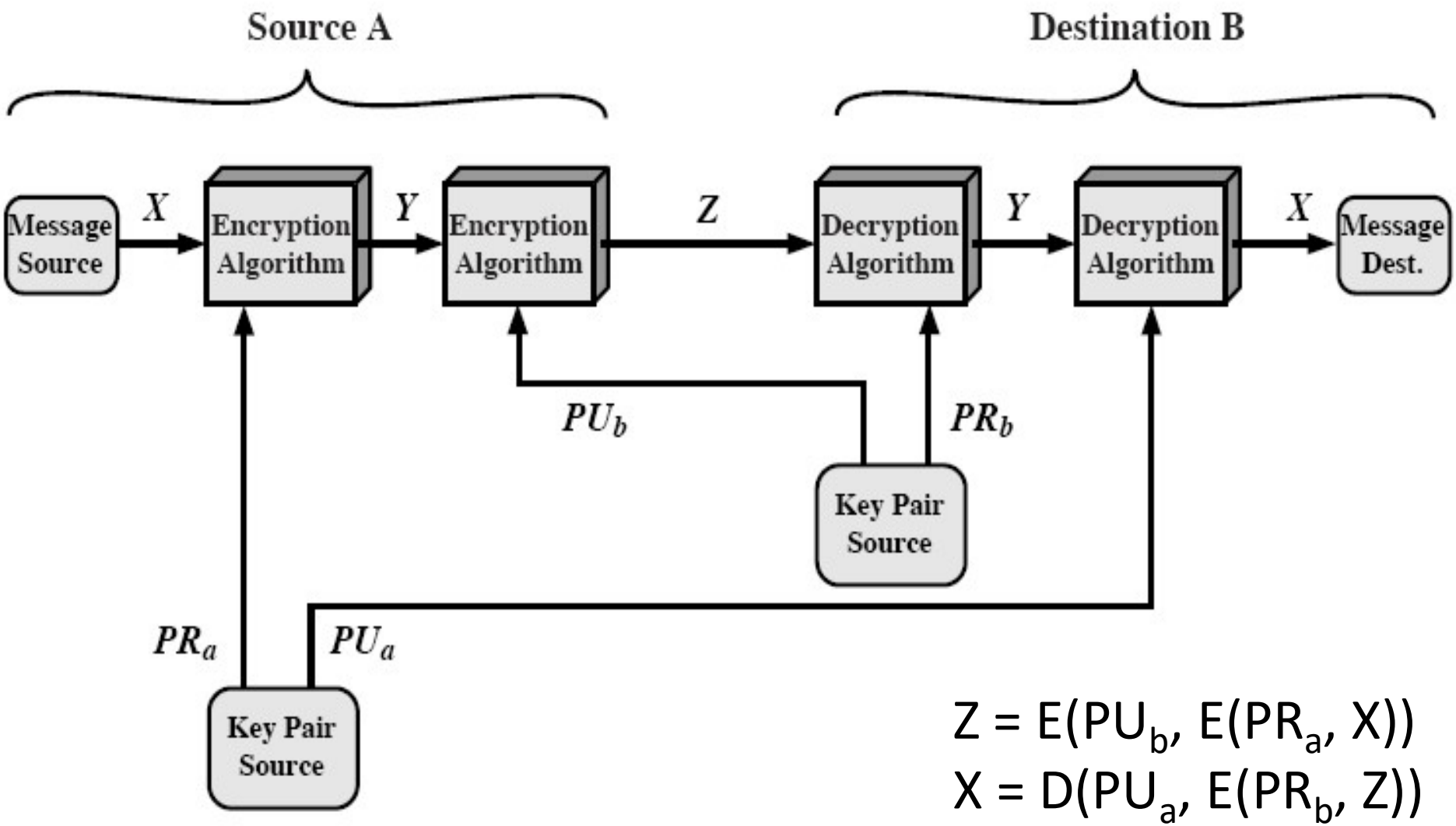
$Y = E(PR_a, X)$

$X = D(PU_a, Y)$

Authentication (cont.)

- A copy also must be stored in ciphertext so that the origin and contents can be verified in case of a dispute.
- A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document.
- Such a block, called an authenticator, must have the property that it is infeasible to change the document without changing the authenticator.
- If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin and content.

Applications: Confidentiality + Authentication



$Z = E(PU_b, E(PR_a, X))$

$X = D(PU_a, E(PR_b, Z))$

Requirements for Public-Key Cryptography (1/2)

1. Computationally easy for a party B to generate a pair (public key PU_b, private key PR_b).
2. Easy for sender to generate ciphertext:

$$C = E(PU_b, M)$$

3. Easy for the receiver to decrypt ciphertext using private key:

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M))]$$

Requirements for Public-Key Cryptography (2/2)

4. Computationally infeasible to determine private key (KR_b) knowing public key (KU_b).
5. Computationally infeasible to recover message M, knowing KU_b and ciphertext C.
6. Either of the two keys can be used for encryption, with the other used for decryption:

$$M = D[PU_b, E(PR_b, M))] = D[PR_b, E(PU_b, M))]$$

RSA

RSA

- 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.
  - nb. exponentiation takes $O((\log n)^3)$ operations (easy)
- Uses large integers (eg. 1024 bits).
- Security due to cost of factoring large numbers.
  - nb. factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA Key Setup

- Each user generates a public/private key pair by:
- Selecting two large primes at random - p, q
- Computing their system modulus $n=p.q$
  - note $\varnothing(n)=(p-1)(q-1)$
  - >512 bits $\rightarrow 1.340e^{154}$
- Selecting at random the encryption key e
    - where $1<e<\varnothing(n)$, $\gcd(e,\varnothing(n))=1$
- Solve following equation to find decryption key d
  - $e.d=1 \bmod \varnothing(n)$ and $0 \leq d \leq n$
- Publish their public encryption key: PU={e,n}
- Keep secret private decryption key: PR={d,n}

RSA Use

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

RSA Example - Key Setup (cont.)

Key Generation by Alice	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Public Key	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

RSA Example - Key Setup

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 = 187$
3. Compute $\varnothing(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e: $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $de=1 \bmod 160$ and $d < 160$ Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$

   Extended Euclidean Method
6. Publish public key $PU=\{7,187\}$
7. Keep secret private key $PR=\{23,187\}$

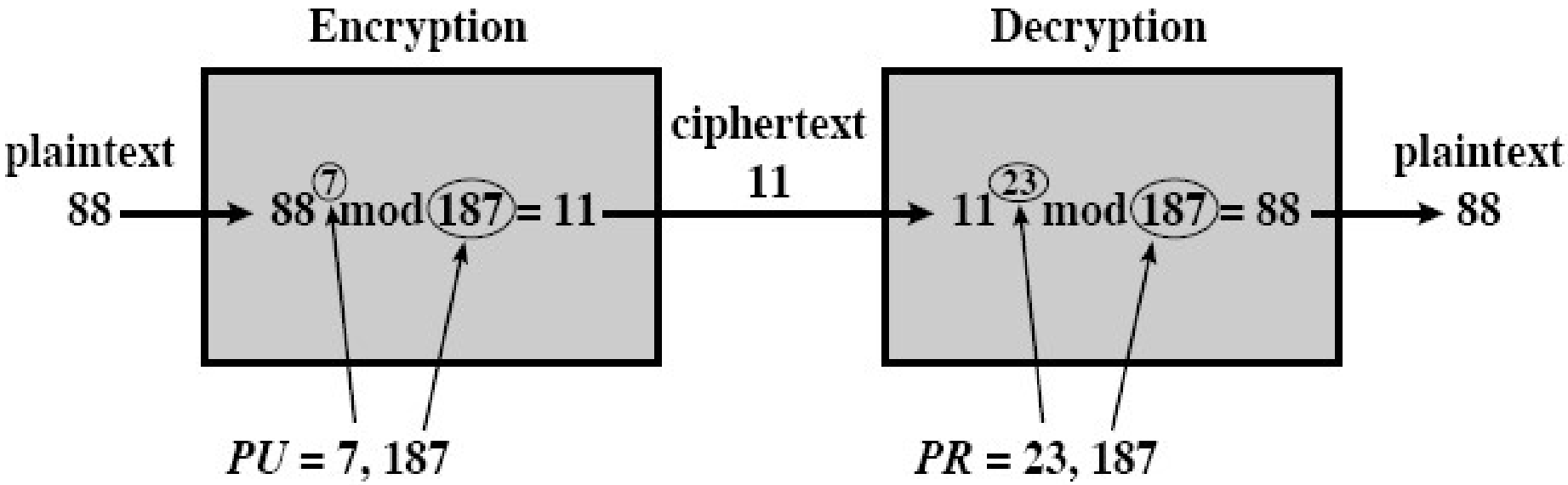
RSA Example - En/Decryption (1/2)

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

  $C = 88^7 \bmod 187 = 11$
- Decryption:

  $M = 11^{23} \bmod 187 = 88$

RSA Example - En/Decryption (2/2)



Exponentiation in Modular Arithmetic- Example 1

- $88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$
- $88^1 \bmod 187 = 88$
- $88^2 \bmod 187 = 7744 \bmod 187 = 77$
- $88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132 = (77 \times 77) \bmod 187$
- $88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

Exponentiation in Modular Arithmetic- Example 2

- $11^{23} \bmod 187 = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$
- $11^1 \bmod 187 = 11$
- $11^2 \bmod 187 = 121$
- $11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$
- $11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33 = 55 \times 55 \bmod 187$
- $11^{23} \bmod 187 = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

Setting up RSA: Another Example

Step 1: Let p = **47** and q = **59**. Thus n = **47** x **59** = **2773**

Step 2: Select e = **17**

Step 3: Publish (n,e) = (**2773**, **17**)

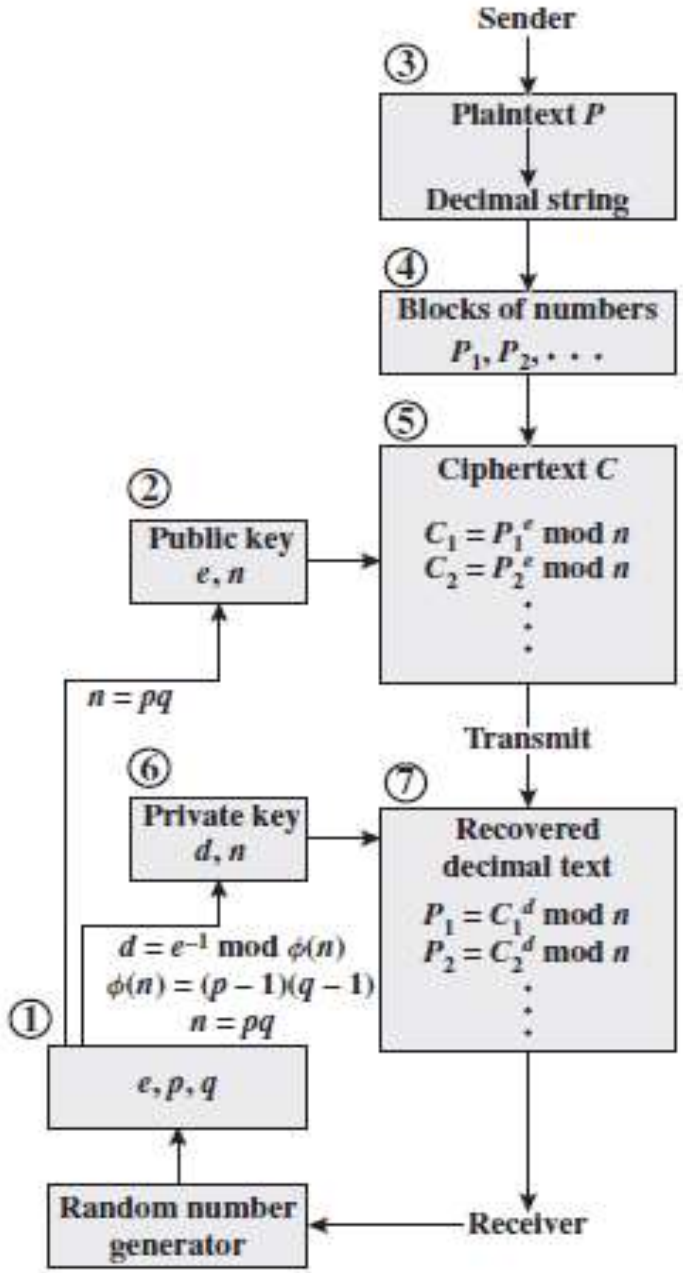
Step 4: (p-1) x (q-1) = **46** x **58** = **2668**

Use the Euclidean Algorithm to compute the modular inverse of **17** modulo **2668**. The result is d = **157**

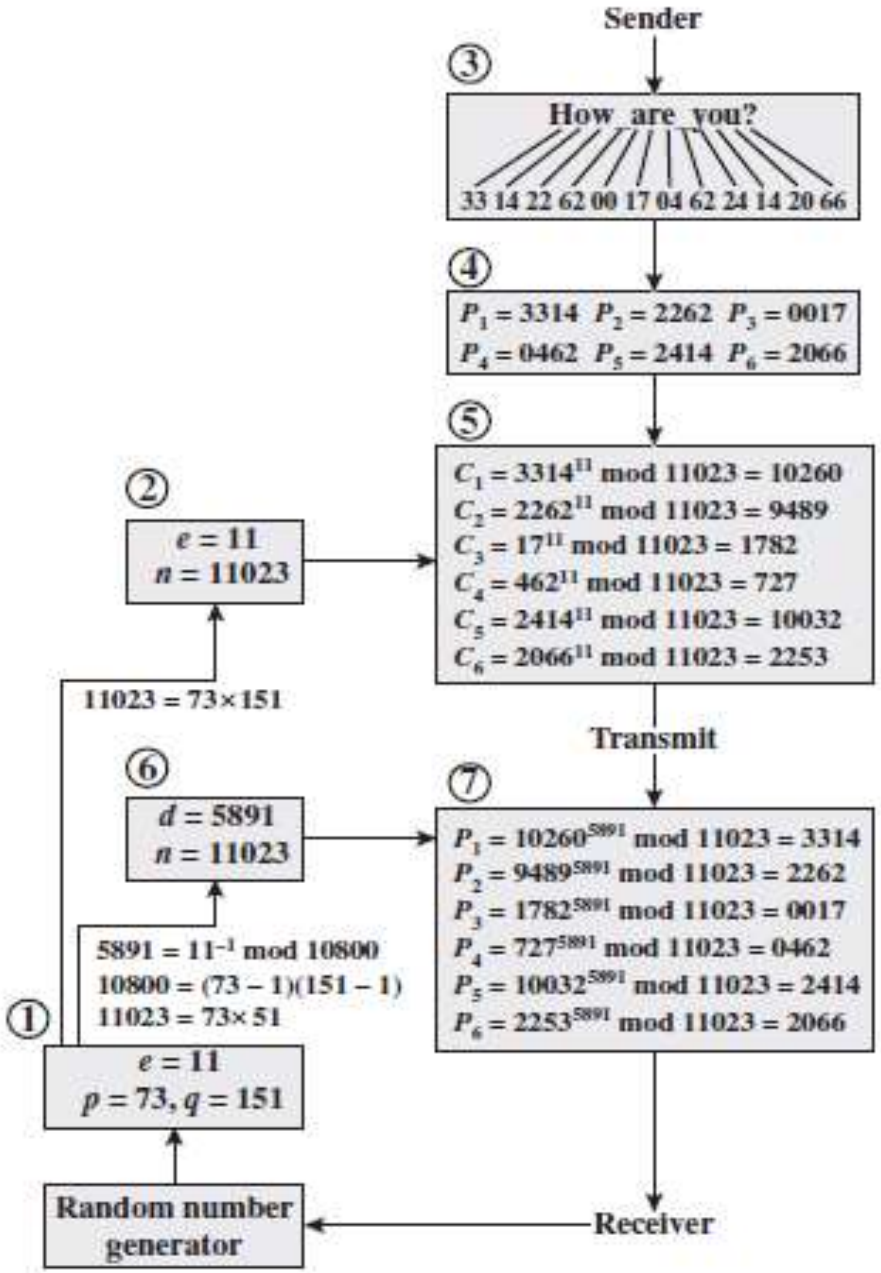
<< Check: **17** x **157** = 2669 = 1(mod **2668**) >>

Public key is (**2773**,**17**)

Private key is **157**



(a) General approach



(b) Example

RSA Security Summary

There are two one-way functions involved in the security of RSA.

One-way function	Description
Encryption function	The encryption function is a trapdoor one-way function, whose trapdoor is the private key. The difficulty of reversing this function without the trapdoor knowledge is **believed** (but not known) to be as difficult as factoring.
Multiplication of two primes	The difficulty of determining an RSA private key from an RSA public key is **known** to be equivalent to factoring n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless they can factor n. Because multiplication of two primes is believed to be a one-way function, determining an RSA private key from an RSA public key is believed to be very difficult.

RSA Security

- Possible approaches to attacking RSA are:
  - Brute force key search (infeasible given size of numbers).
  - Mathematical attacks (based on difficulty of computing ø(n), by factoring modulus n).
  - Timing attacks (on running of decryption).