Public key Cryptography

- **Public key cryptography** (PKC) is an **encryption** technique that uses a paired **public** and private **key** algorithm for secure data communication.
- A message sender uses a recipient's **public key** to encrypt a message.
- To decrypt the sender's message, only the recipient's **private key** may be used.

Principles of Public-Key Cryptosystems

- The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption:

Principles of Public-Key Cryptosystems

Key distribution

- The communicants already shares a key or someone has been distributed the key.
- How to secure communications in general without having to trust a KDC with your key

Digital signatures

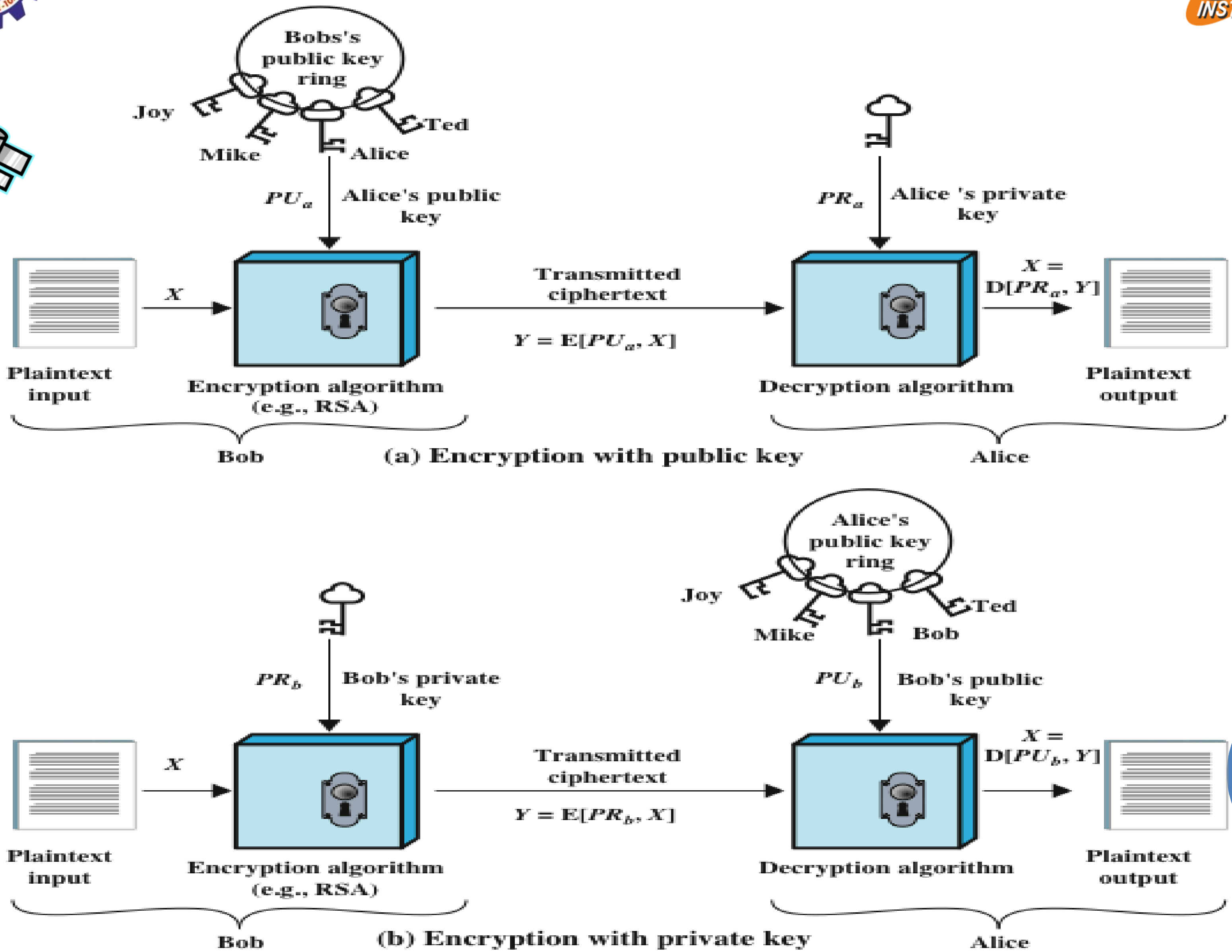
- How to verify that a message comes intact from the claimed sender

Principles of Public-Key Cryptosystems

A public-key encryption scheme has six ingredients

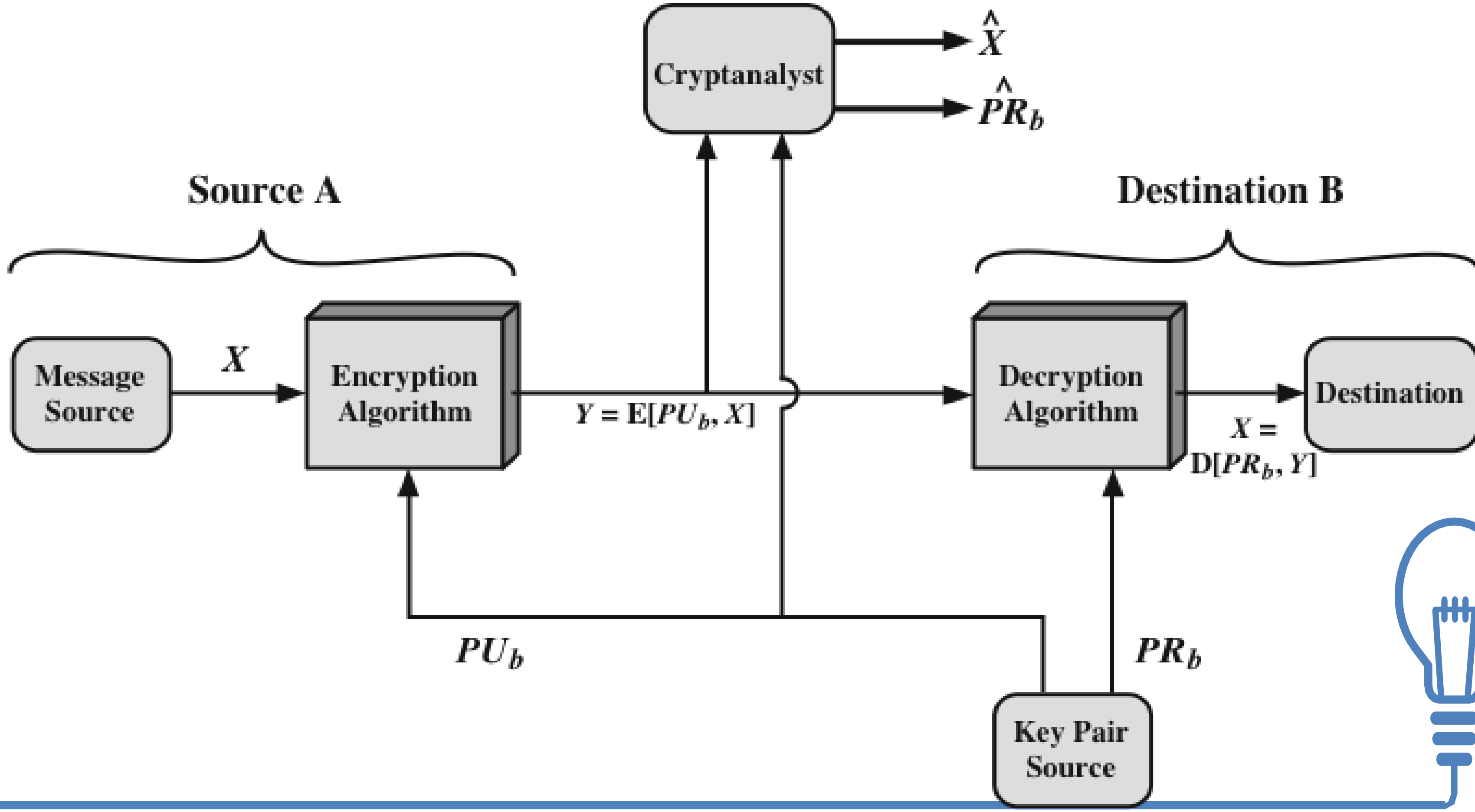
Plaintext	Encryption algorithm	Public key	Private key	Ciphertext	Decryption algorithm
The readable message or data that is fed into the algorithm as input	Performs various transform-ations on the plaintext	Used for encryption or decryption	Used for encryption or decryption	The scrambled message produced as output	Accepts the ciphertext and the matching key and produces the original plaintext

Principles of Public-Key Cryptosystems



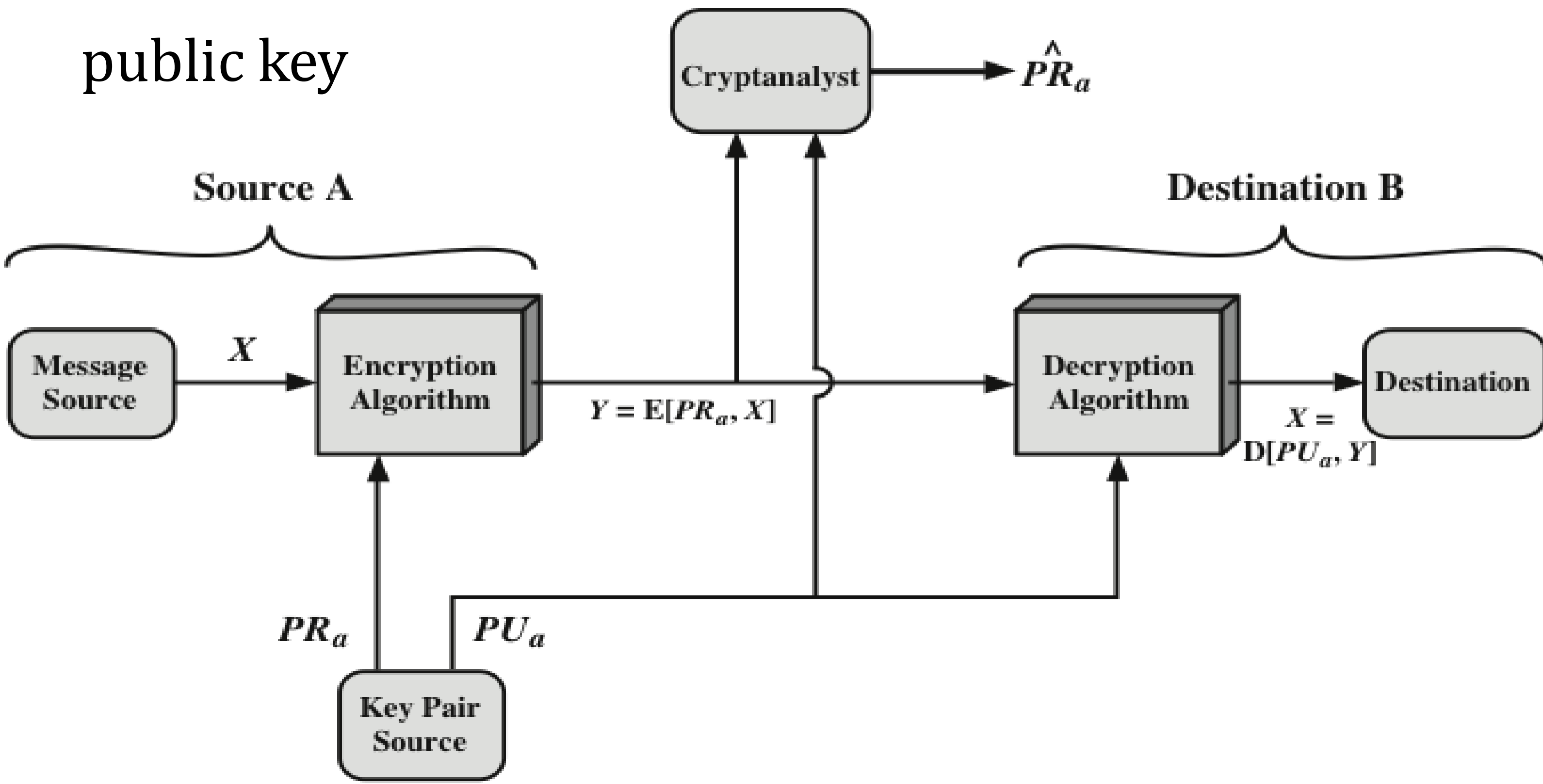
Public-Key Cryptosystem:
Encryption using public key -Secrecy

This figure provides confidentiality because two related key used for encryption other being used for decryption



Public-Key Cryptosystem: Encryption using private key -Authentication

There is no protection of confidentiality because any observer can decrypt the message by using the sender's public key

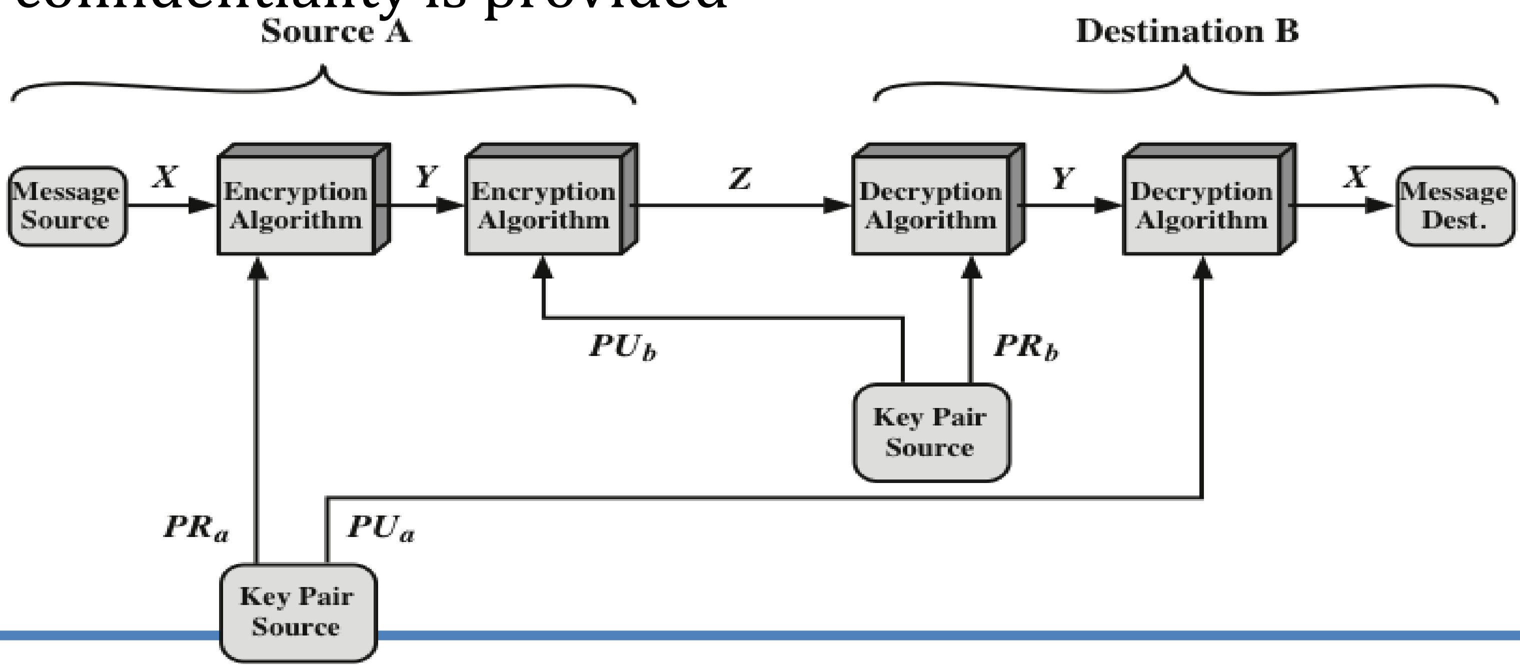


Public-Key Cryptosystem: Authentication and Secrecy

Encrypting a message, using the sender's private key. This provides the digital signature.

Next, encrypt again, using the receiver's public key.

The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus, confidentiality is provided



Applications for Public-Key Cryptosystems

- Public-key cryptosystems can be classified into three

Encryption/decryption

- The sender encrypts a message with the recipient's public key

Digital signature

- The sender "signs" a message with its private key

Key exchange

- Two sides cooperate to exchange a session key

Applications for Public-Key Cryptosystems

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

Conditions that these algorithms must fulfil:

It is computationally easy

- for a party B to generate a pair (public- key PU_b, private key PR_b)
- for a sender A, knowing the public key and the message to be encrypted, to generate the corresponding ciphertext
- for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message

It is computationally infeasible

- for an adversary, knowing the public key, to determine the private key.
- for an adversary, knowing the public key and a ciphertext, to recover the original message.

The two keys can be applied in either order.

Rivest-Shamir-Adleman (RSA) Scheme

- RSA is the algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm.
- Asymmetric means that there are two different keys. This is also called public key cryptography, because one of them can be given to everyone. The other key must be kept private.
- One of the first successful responses to the challenge was Developed in 1977 at MIT by Ron Rivest, Adi Shamir & Len Adleman

RSA Algorithm

- Plaintext is encrypted in blocks with each block having a binary value less than some number *n*
- Encryption and decryption are of the following form, for some plaintext block *M* and cipher text block C
  - $C = M^e \bmod n$
  - $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$
- Both sender and receiver must know the value of *n*
- The sender knows the value of *e,* and only the receiver knows the value of *d*
- This is a public-key encryption algorithm with a public key of $PU=\{e,n\}$ and a private key of $PR=\{d,n\}$

Algorithm Requirements

- For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

1. It is possible to find values of *e, d, n* such that $M^{ed} \bmod n = M$ for all $M<n$
2. It is relatively easy to calculate $M^e \bmod n$ and $C^d \bmod n$ for all values of $M < n$
3. It is infeasible to determine *d* given *e* and *n*

Key Generation by Alice	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \pmod{\phi(n)}$
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 Private Key	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Example of RSA Algorithm

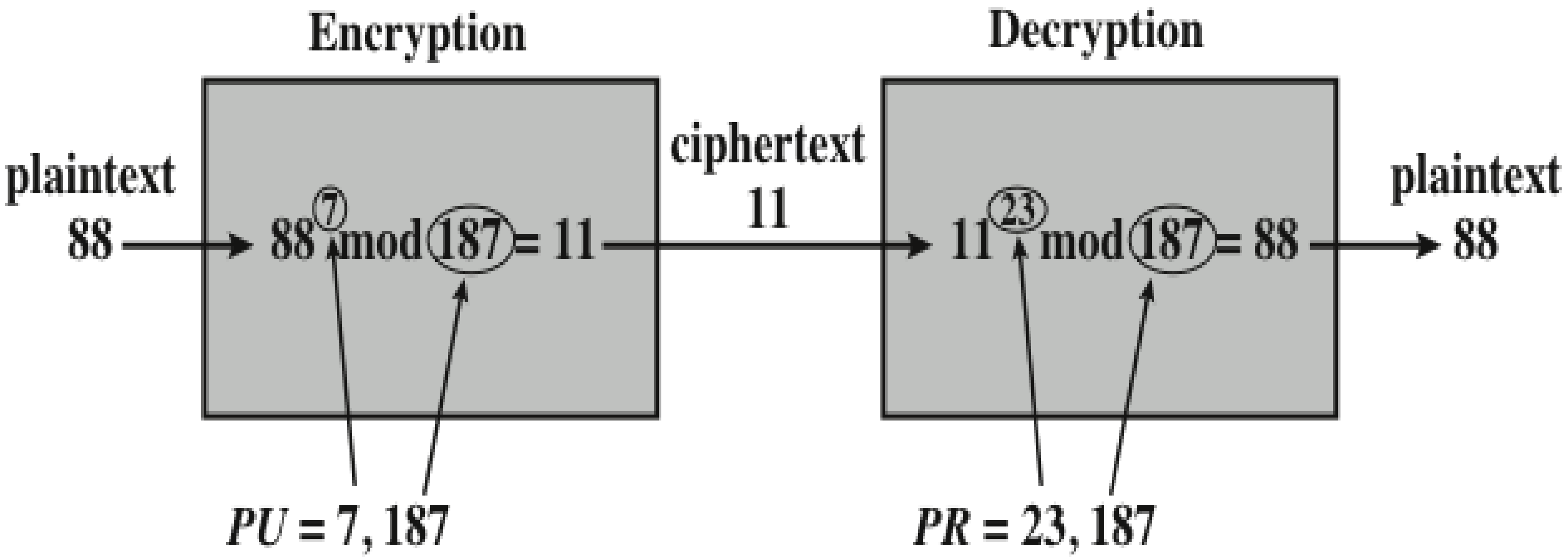
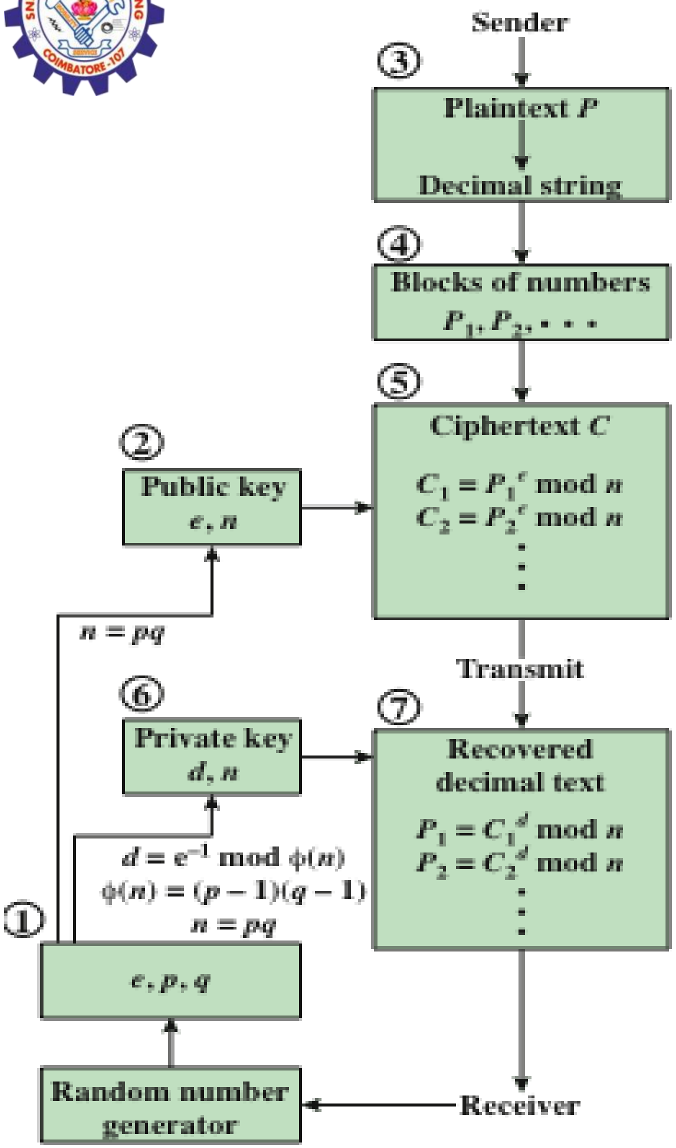
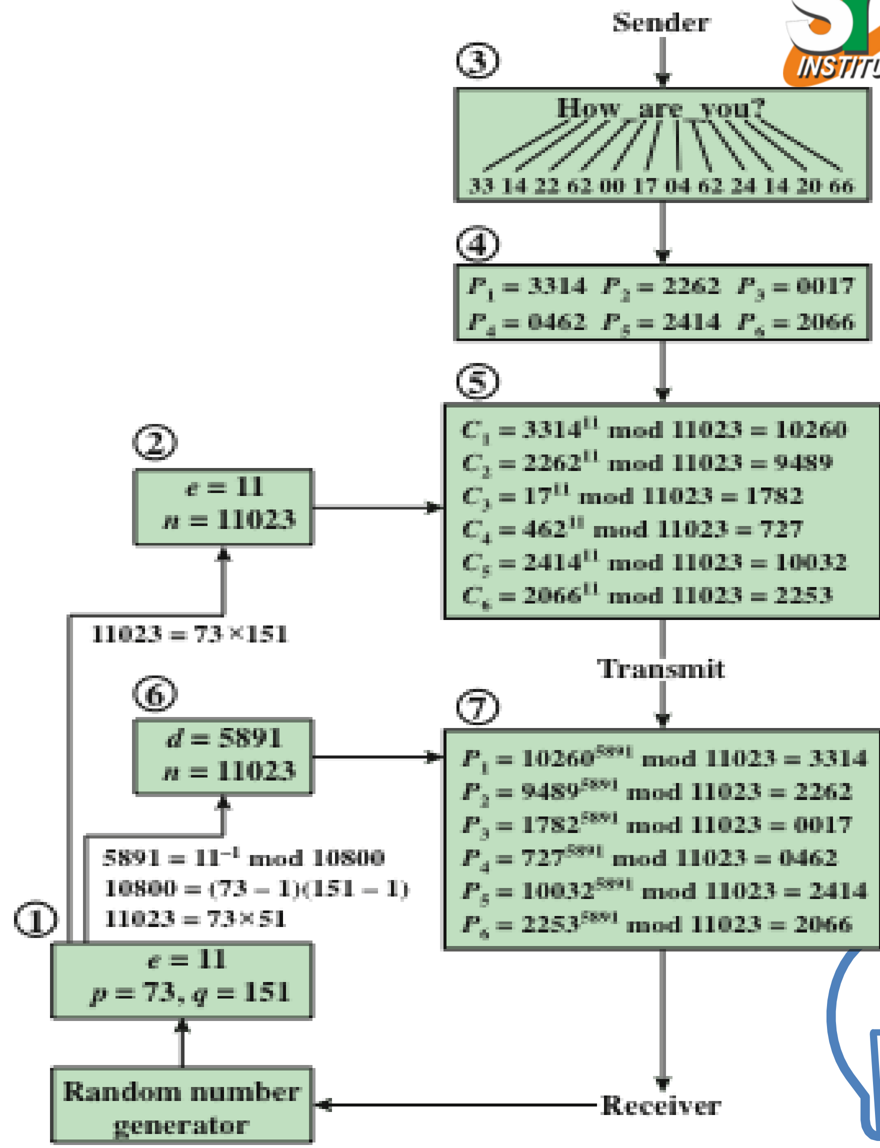


Figure 9.6 Example of RSA Algorithm



(a) General approach



(b) Example

The Security of RSA

Five possible approaches to attacking RSA are:

Chosen ciphertext attacks

- This type of attack exploits properties of the RSA algorithm

Brute force

- Involves trying all possible private keys

Mathematical attacks

- There are several approaches, all equivalent in effort to factoring the product of two primes

Hardware fault-based attack

- This involves inducing hardware faults in the processor that is generating digital signatures

Timing attacks

- These depend on the running time of the decryption algorithm

THANK YOU!!!