Lecture 8:

Public-Key Cryptography and RSA

4th Year- Course, CCSIT, UoA

Lecture Goals

1. To review public-key cryptography
2. To demonstrate that confidentiality and sender-authentication can be achieved simultaneously with public-key cryptography
3. To review the Rivest-Shamir-Adleman (RSA) algorithm for public-key cryptography
4. To discuss the security of RSA

Public-key Cryptography

- Public-key cryptography is also known as asymmetric-key cryptography.
- Encryption and decryption is carried out using ***two different keys.*** The two keys in such a key pair are referred to as the public key and the private key.
- This solves one of the most vexing problems associated with symmetric-key cryptography — the problem of key distribution.
- With public key cryptography, all parties interested in secure communications can publish their public keys.

Providing Confidentiality

- Public-key cryptography can be used to provide confidentiality (or secrecy) for a message.
- Party A, if wanting to communicate confidentially with party B, can encrypt a message using B's publicly available key. Such a communication would only be decipherable by B as only B would have access to the corresponding private key.
- This is illustrated in he next Figure.
- The notation here is that; A 's public and private keys are designated PU_a and PR_a. B 's public and private keys are designated PU_b and PR_b, respectively.

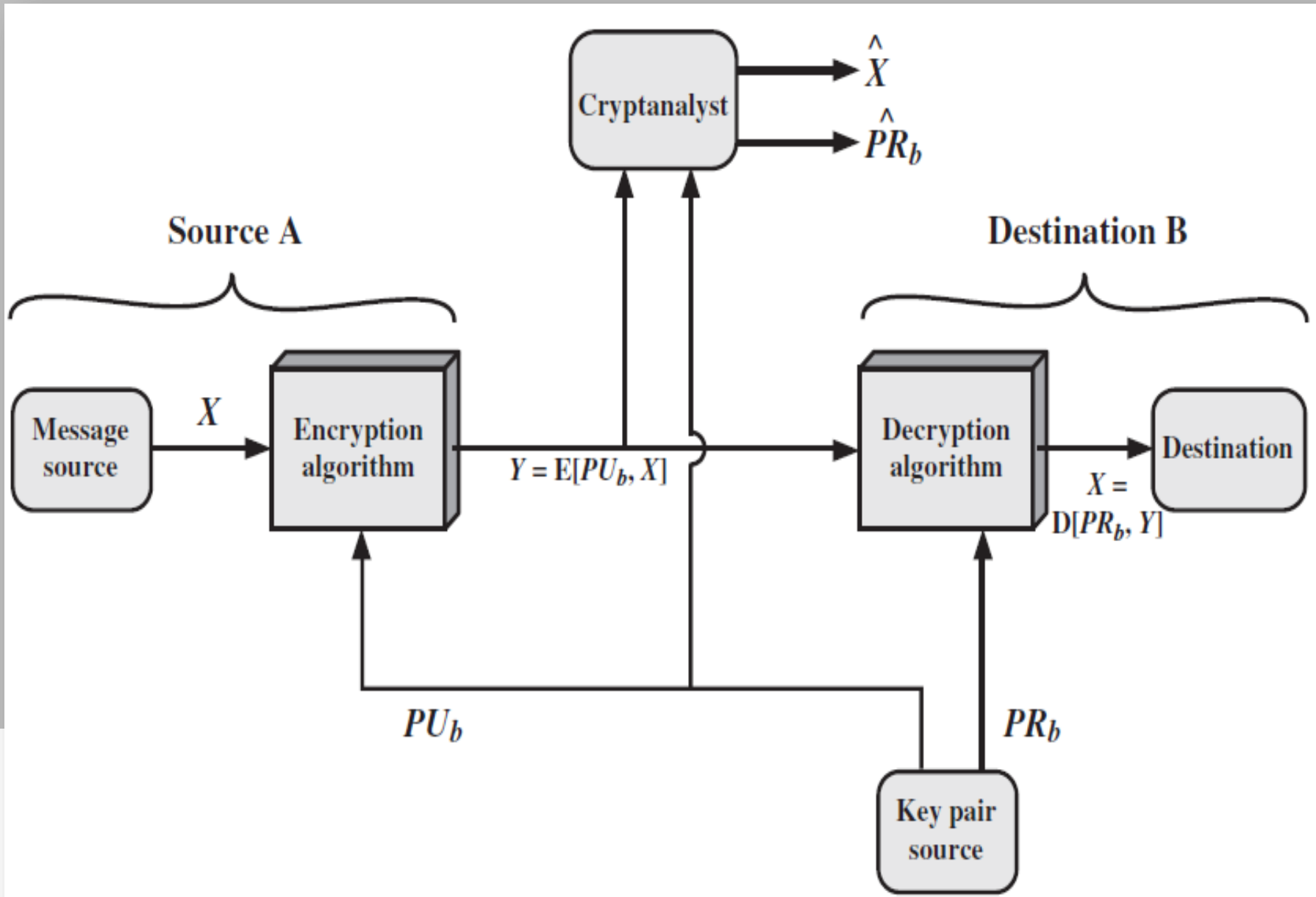

Figure 9.2 Public-Key Cryptosystem: Secrecy

Providing Authentication

- Public-key cryptography can also be used to provide authentication.
- Party A, if wanting to send an authenticated message to party B, would encrypt the message with A's own private key.
- Since this message would only be decipherable with A's public key, that would establish the authenticity of the message — meaning that A was indeed the source of the message.
- This is illustrated in the next Figure.

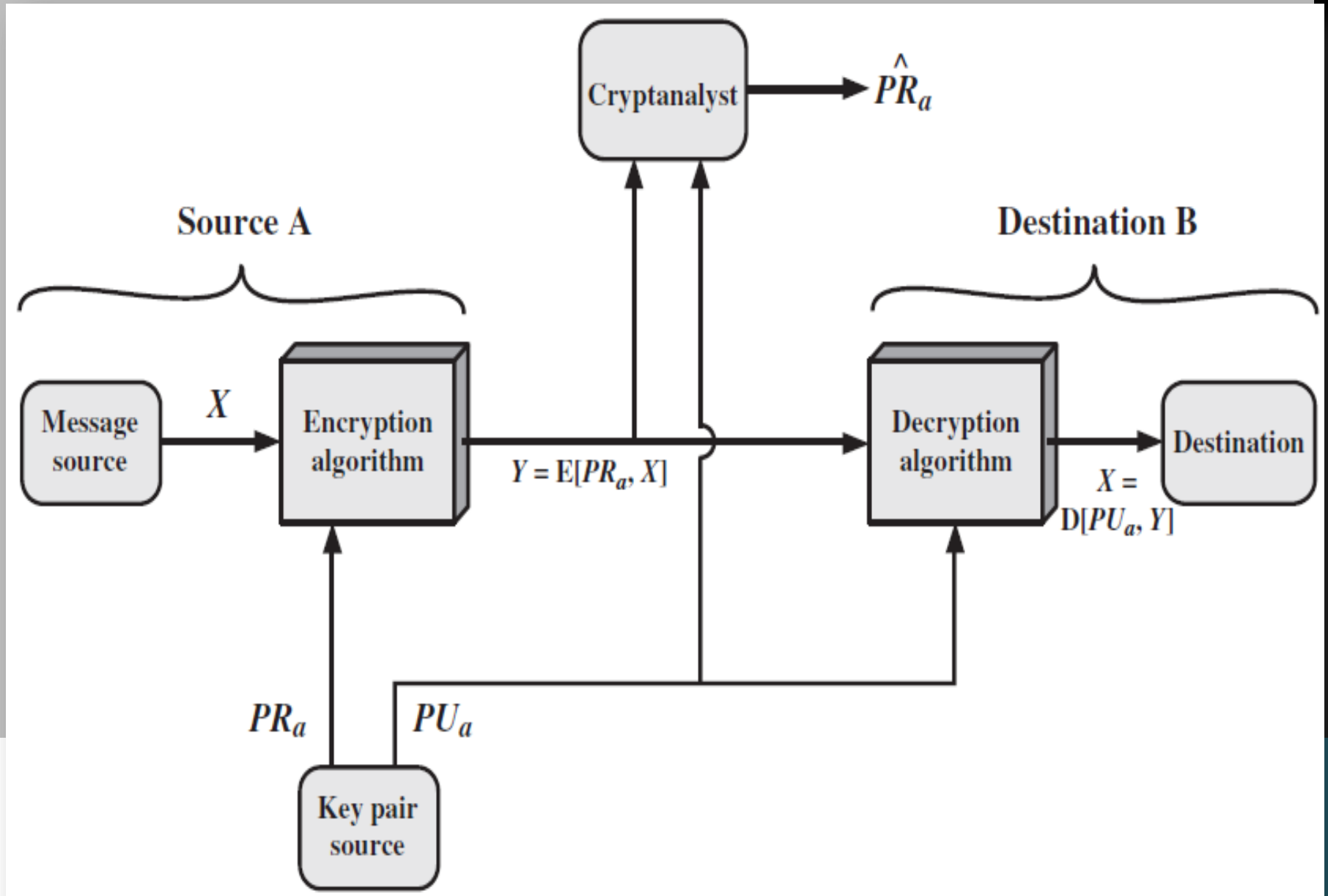

Figure 9.3 Public-Key Cryptosystem: Authentication

Both Confidentiality and Authentication (1)

- ✓ The communication link in the next Figure shows how public-key encryption can be used to provide both confidentiality and authentication at the same time.
- ✓ Note again that confidentiality means that we want to protect a message from eavesdroppers and authentication means that the recipient needs a guarantee as to the identity of the sender.
- ✓ Let's say that A wants to send a message M to B with both authentication and confidentiality. The processing steps undertaken by A to convert M into its encrypted form C that can be placed on the wire are:

$$C = E(P U_b, E(P R_a, M))$$

where $E(\ )$ stands for encryption.

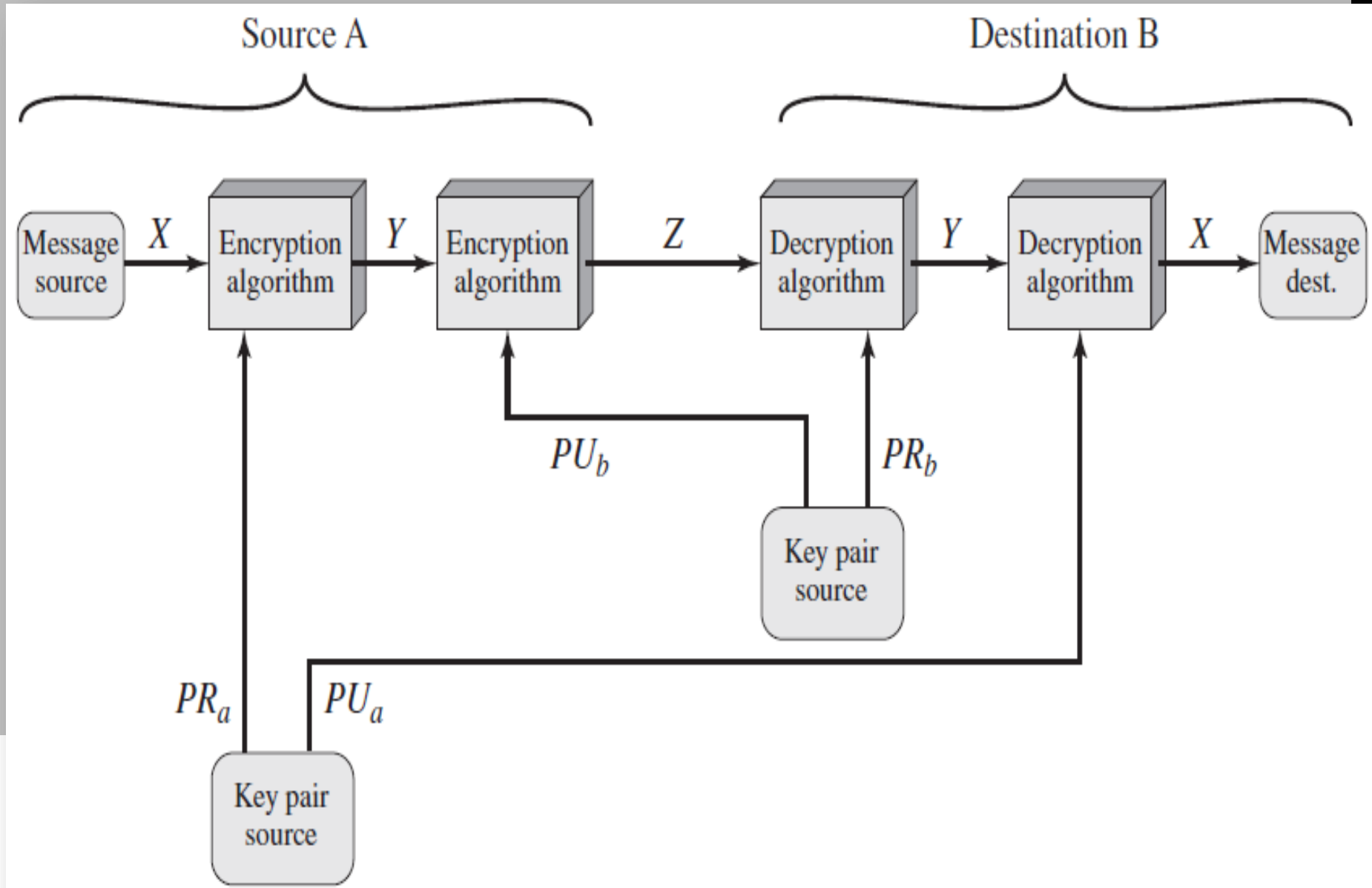

Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Both Confidentiality and Authentication (2)

- ✓ The processing steps undertaken by B to recover M from C are $M = D(P U_a, D(P R_b, C))$

  where $D( )$ stands for decryption.

- ✓ The sender A encrypting his/her message with its own private key $P R_a$ provides authentication. This step constitutes A putting his/her **digital signature** on the message.

- ✓ Instead of applying the private key to the entire message, a sender may also "sign" a message by applying his/her private key to just a **small block** of data that is derived from the message to be sent.

- ✓ The sender A further encrypting his/her message with the receiver's public key $P U_b$ provides **confidentiality**.

More on public-key cryptography

- The price paid for achieving confidentiality and authentication at the same time is that now the message must be processed four times in all for encryption/ decryption.
- Because of the greater computational overhead associated with public-key cryptosystems, public-key cryptography does not make obsolete the more traditional symmetric-key cryptography.
- However, it is generally agreed that public-key encryption is indispensable for key management and digital signature applications.

The RSA Cryptosystem

The RSA scheme is one of the most widely applied public-key systems. The basic computational steps for key generation in RSA are:

1. *Generate two different* ***primes*** p *and* q
2. *Calculate the modulus* $n = p \times q$
3. *Calculate the totient function* $\phi(n) = (p - 1) \times (q - 1)$
4. *Select integer* e *such that* $1 < e < \phi(n)$, $gcd(\phi(n), e) = 1$
5. *Calculate* d *such that* $d = e^{-1} \bmod \phi(n)$
6. *Public Key* $= [e, n]$
7. *Private Key* $= [d, n]$

The RSA Algorithm (1)

- Now suppose we are given an integer M, $M < n$, that represents our message, then we can transform M into another integer C that will represent our ciphertext by the following ***modular exponentiation***:

$$C = M^e \bmod n \qquad \textbf{(encryption)}$$

- As you will soon see, we can recover M back from C by the following modulo operation

$$M = C^d \bmod n \qquad \textbf{(decryption)}$$

- An individual who wishes to receive messages confidentially will use the pair of integers $\{e, n\}$ has his/her public key.
- At the same time, this individual can use the pair of integers $\{d, n\}$ as the private key.

The RSA Algorithm (2)

- RSA could be used as a block cipher for the encryption of the message. The block size would equal the number of bits required to represent the modulus n .
- If the modulus required, say, 1024 bits for its representation, message encryption would be based on 1024-bit blocks. Every plaintext block can now be thought of as an integer M of value

  $$0 \leq M \leq 2^{1024} - 1.$$

- If this integer is expressed in decimal form, its value would be less than 10^{309}. In other words, the message integer M will have 309 decimal digits for each block of the plaintext.