Computer Systems Security

Dr. Ayman Abdel-Hamid

College of Computing and Information Technology

Arab Academy for Science & Technology and
Maritime Transport

Chapter 9

Public-Key Cryptography and RSA

Outline

- Principles of Public-Key Cryptosystems
- RSA Algorithm

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

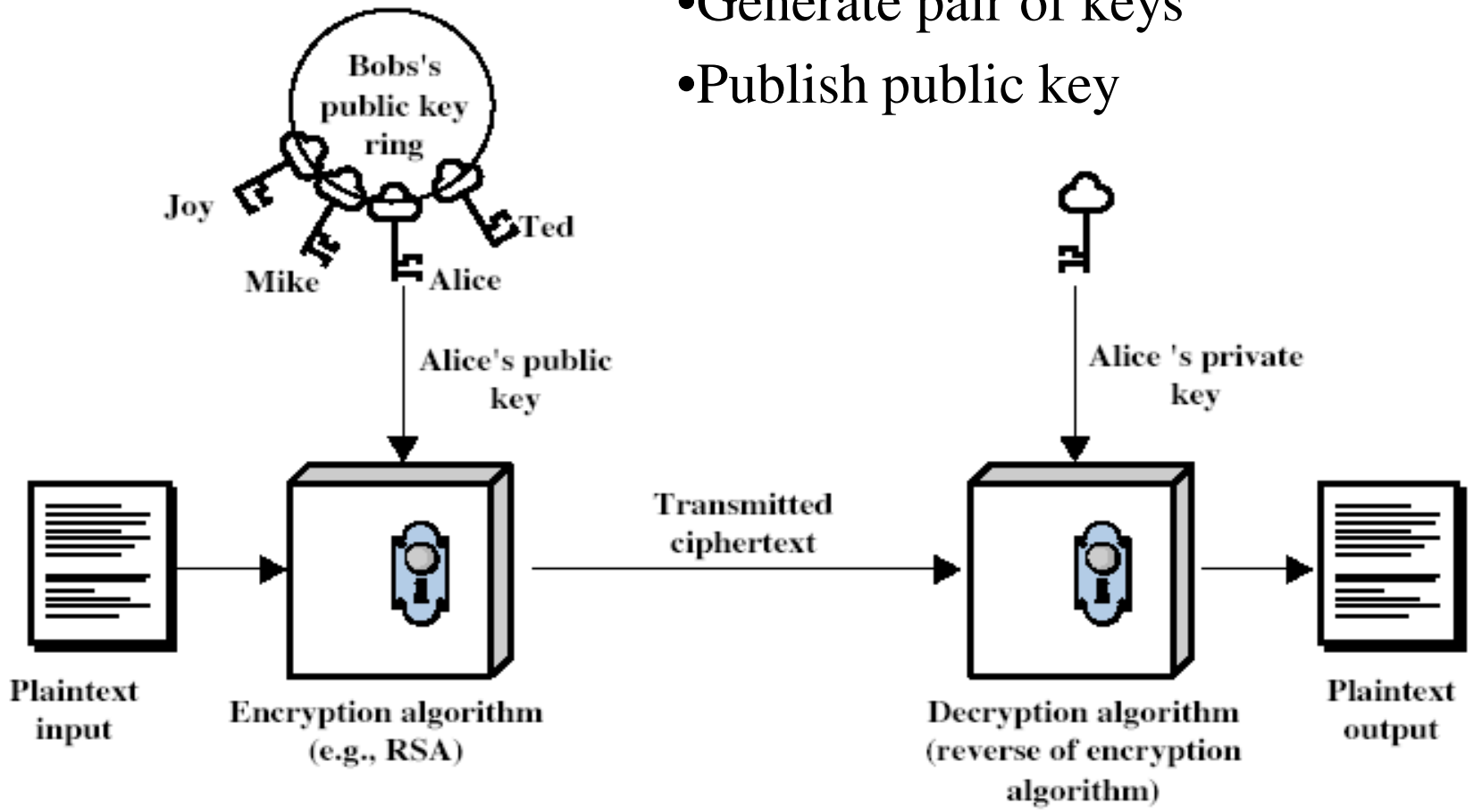
- probably most significant advance in the 3000 year history of cryptography
- uses **two** keys – a public & a private key
- **asymmetric** since parties are **not** equal
- uses clever application of number theoretic concepts to function
- complements **rather than** replaces private key crypto

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

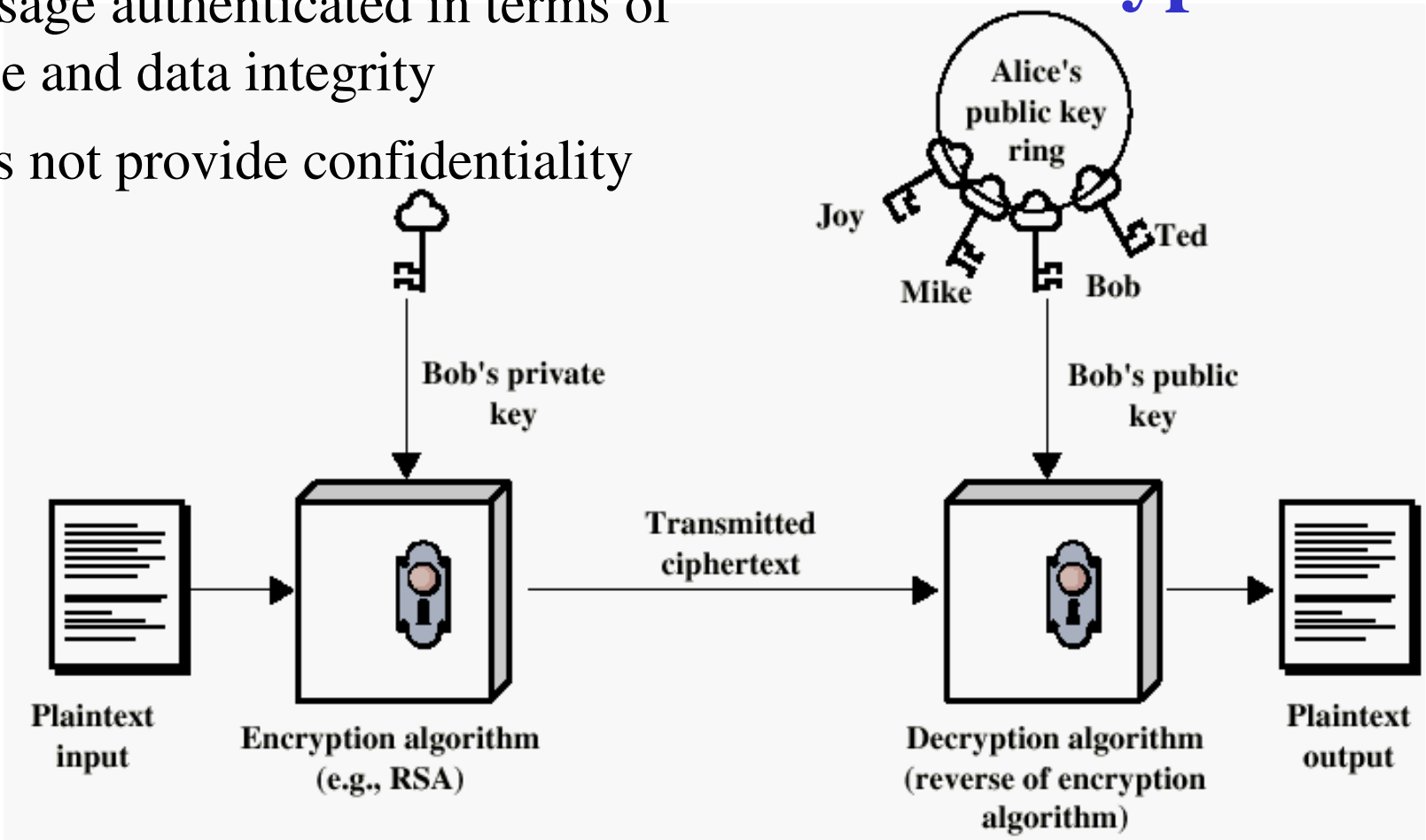
Public-Key Cryptography: Confidentiality

- Generate pair of keys
- Publish public key



Authentication using Public-Key Crypto

- Entire encrypted message serves as a DS (can encrypt some bits as authenticator)
- Message authenticated in terms of source and data integrity
- Does not provide confidentiality



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 invention due to *Whitfield Diffie* & *Martin Hellman* at Stanford Univ. in 1976
  - known earlier in classified community

Public-Key Characteristics

- Public-Key algorithms rely on two keys with the characteristics that it is:
  - computationally infeasible to find decryption key knowing only algorithm & encryption key
  - 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 (in some schemes)

Public-Key Cryptosystems

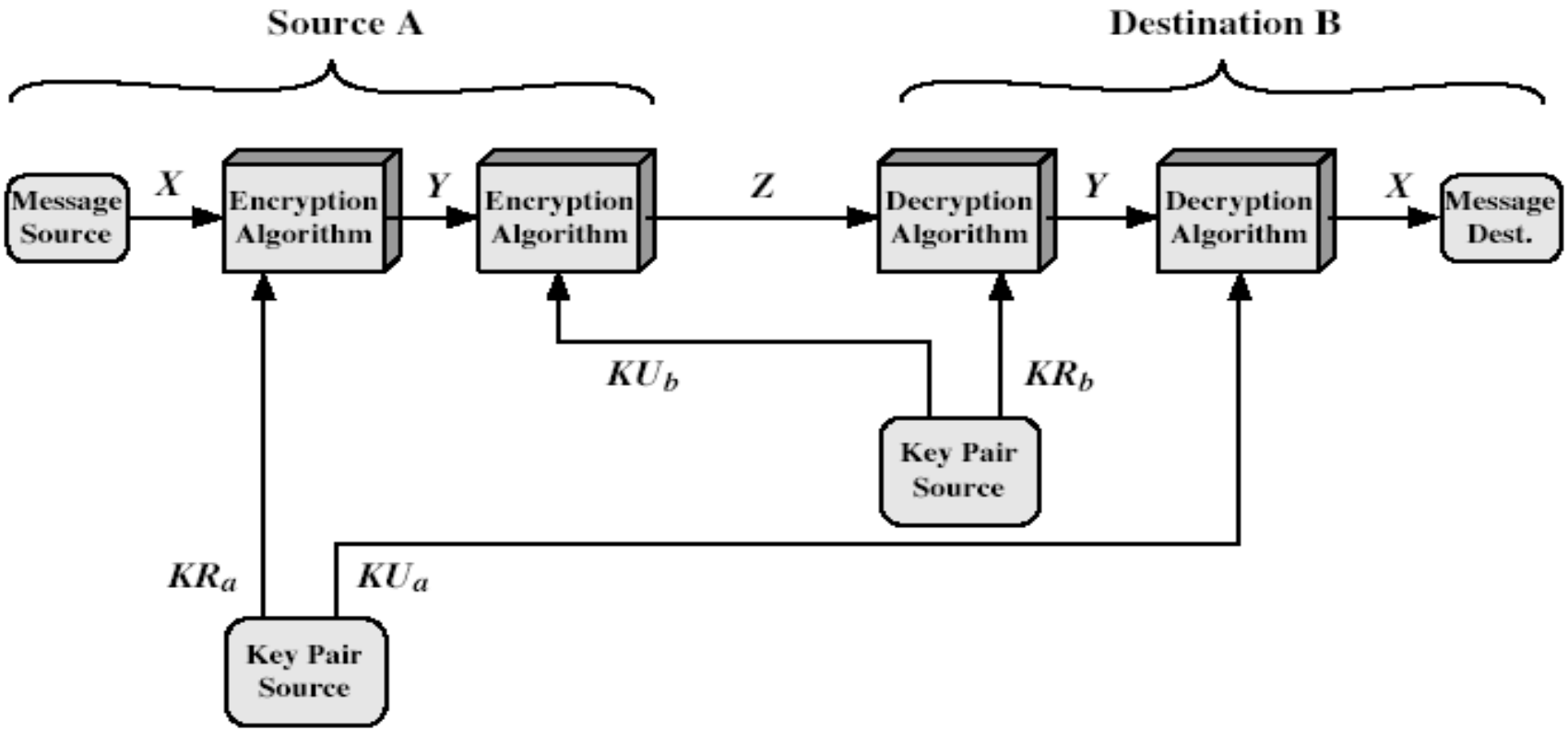


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy → sender encrypts a message with the recipient's public key)
  - **digital signatures** (provide authentication → sender signs a message with its private key)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Requirements for Public-Key Crypto

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

$$C = E_{KUb}(M)$$

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

$$M = D_{KRb}(C) = D_{KRb}[E_{KUb}(M)]$$

Requirements for Public-Key Crypto

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. Encryption and decryptions functions can be applied in either order

$$M = D_{KRb}[E_{KUb}(M)] = D_{KUb}[E_{KRb}(M)]$$

Security of Public Key Schemes

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalysis) problems
- requires the use of **very large numbers**
- hence is **slow** compared to secret key schemes
- *Public-key encryption currently confined to key management and signature applications*

RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- Block cipher (use large numbers n = 1024 bits)
- For plaintext block M and ciphertext block C
  - C = M^e mod n
  - M = C^d mod n
  - Sender and receiver know n
  - Sender knows e
  - Receiver knows d
  - Public key $KU = \{e,n\}$
  - Private key $KR = \{d,n\}$

RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random - p, q
- computing their system modulus $n=p.q$ (factorization of large numbers)
  - note $\varnothing(n)=(p-1)(q-1)$
- 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 $0 \le d \le n$
- publish their public encryption key: KU={e,n}
- keep secret private decryption key: KR={d,p,q}

RSA Use

- to encrypt a message M, the sender:
  - obtains **public key** of recipient $KU=\{e,n\}$
  - computes: $C=M^e \bmod n$, where $0 \leq M < n$
- to decrypt the ciphertext C, the receiver:
  - uses their private key $KR=\{d,p,q\}$
  - computes: $M=C^d \bmod n$
- note that the message M must be smaller than the modulus *n* (block if needed)

RSA Example

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 = 187$
3. Compute $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e: $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $d.e=1 \bmod 160$ and $d < 160$
   Value is $d=23$ since $23 \times 7 = 161 = 1 \times 160 + 1$
6. Publish public key $KU=\{7,187\}$
7. Keep secret private key $KR=\{23,17,11\}$

RSA Example cont.

- sample RSA encryption/decryption is:
- given message $M = 88$ (note that $88<187$)
- encryption:

  $C = 88^7 \bmod 187 = 11$

- decryption:

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