Public Key Cryptography

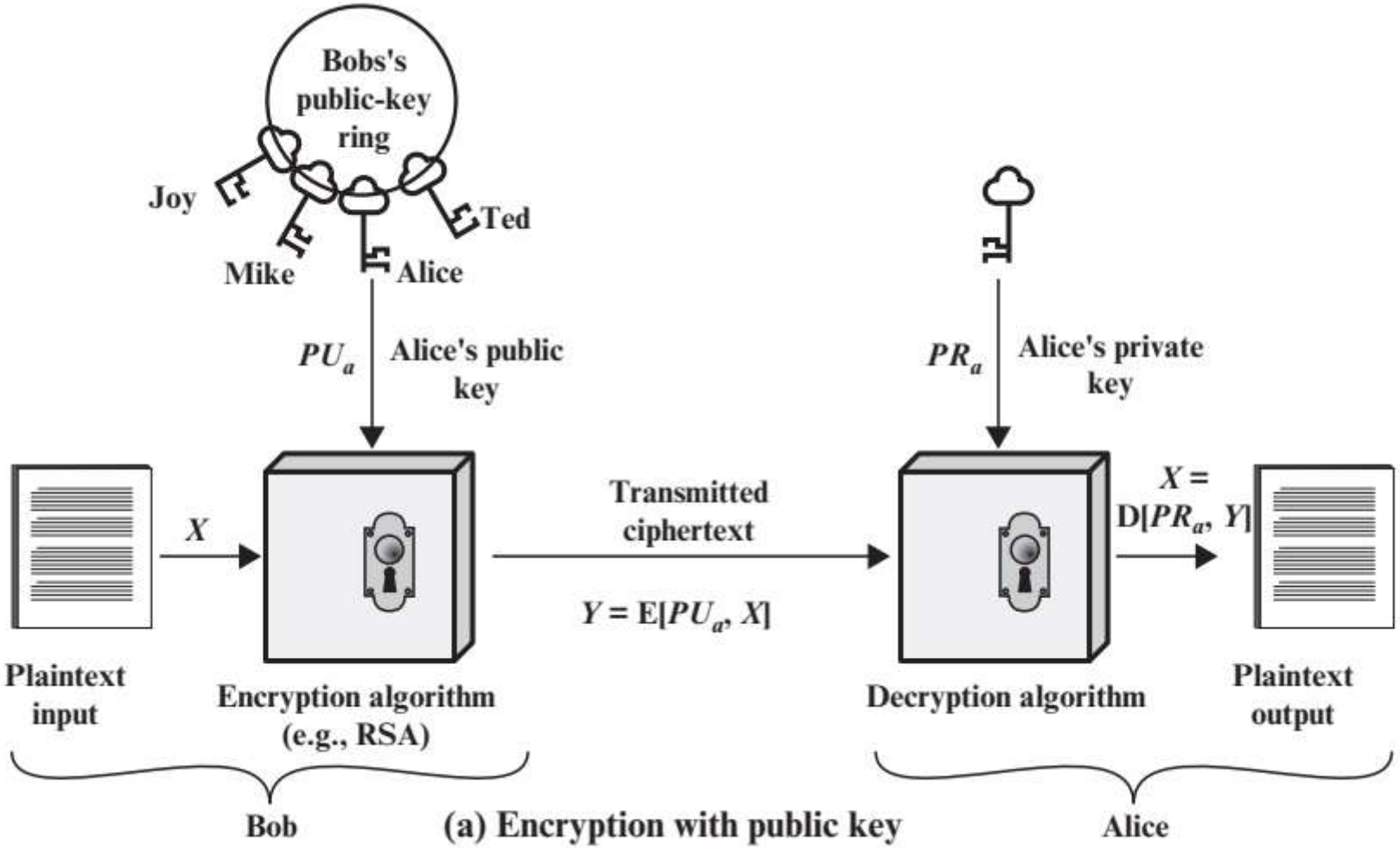
Jyoti Maheshwari

Principles of Public Key Cryptosystems

- Public-key, or asymmetric, cryptography provides a radical departure from all that has gone before. For one thing, public-key algorithms are based on mathematical functions rather than on substitution and permutation.
- public-key cryptography is asymmetric, involving the use of two separate keys, in contrast to symmetric encryption, which uses only one key.
- There is nothing in principle about either symmetric or public-key encryption that makes one superior to another from the point of view of resisting cryptanalysis.
- Because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that symmetric encryption will be abandoned.
- Finally, there is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for symmetric encryption. In fact, some form of protocol is needed, generally involving a central agent, and the procedures involved are not simpler nor any more efficient than those required for symmetric encryption
- theory of public-key cryptosystems is based on number theory

Principles of Public Key Cryptosystems

- Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristic.
  - It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.
- In addition, some algorithms, such as RSA, also exhibit the following characteristic.
  - Either of the two related keys can be used for encryption, with the other used for decryption.
- A **public-key encryption** scheme has six ingredients
- **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext
- **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
- **Ciphertext:** This is the encrypted message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
- **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.



(a) Encryption with public key

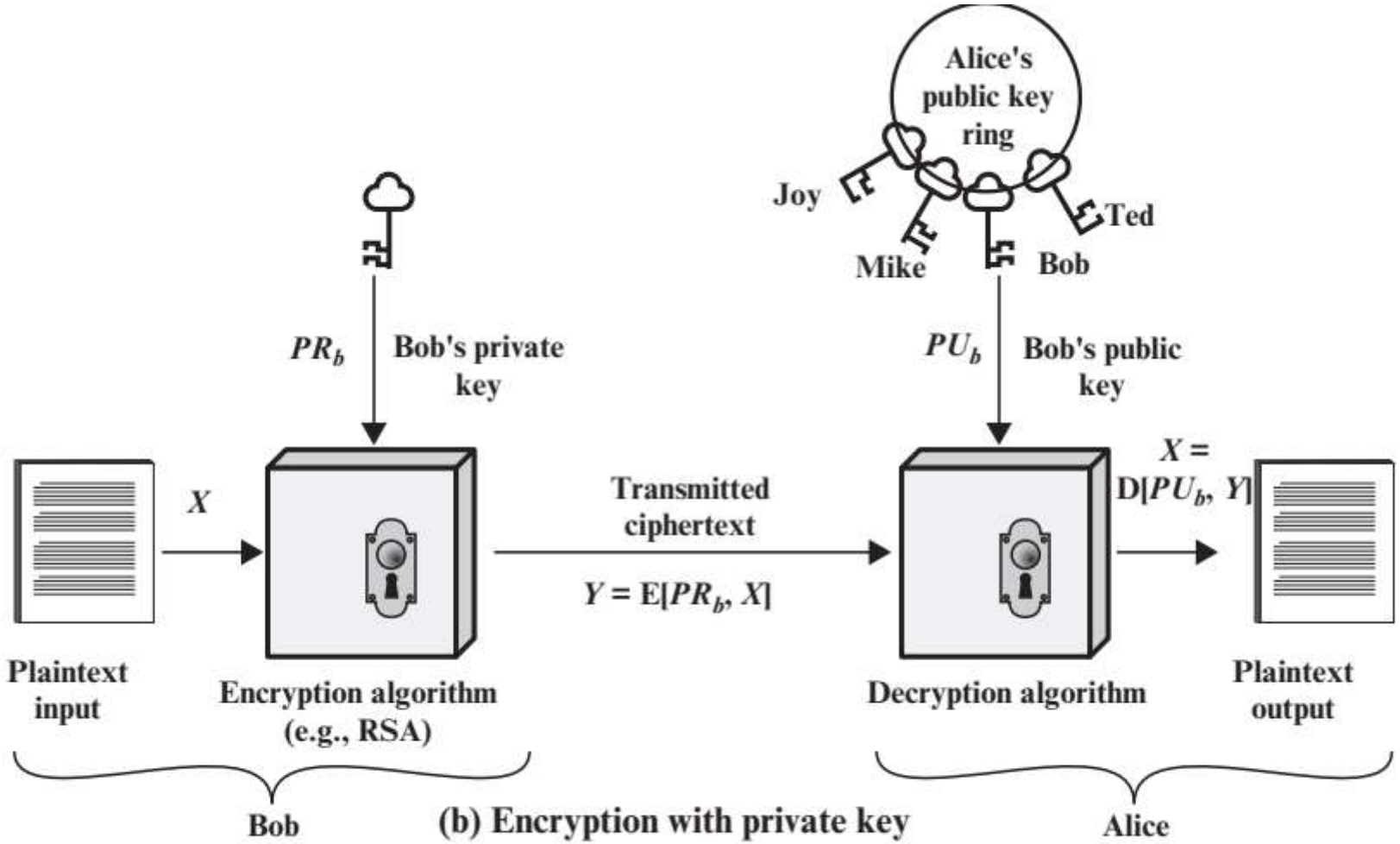


Figure 9.1 Public-Key Cryptography

Essential steps

- The essential steps are the following.
- Each user generates a pair of keys to be used for the encryption and decryption of messages.

- Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private.
  Each user maintains a collection of public keys obtained from others.

- If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

- When Alice receives the message, she decrypts it using her private key.

  No other recipient can decrypt the message because only Alice knows Alice's private key.

As long as a user's private key remains protected and secret, incoming communication is secure. At any time, a system can change its private key and publish the companion public key to replace its old public key.

To discriminate between the two, we refer to the key used in symmetric encryption as a **secret key**. The two keys used for asymmetric encryption are referred to as the **public key** and the **private key**.

Table 9.2 Conventional and Public-Key Encryption

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the key is kept secret. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

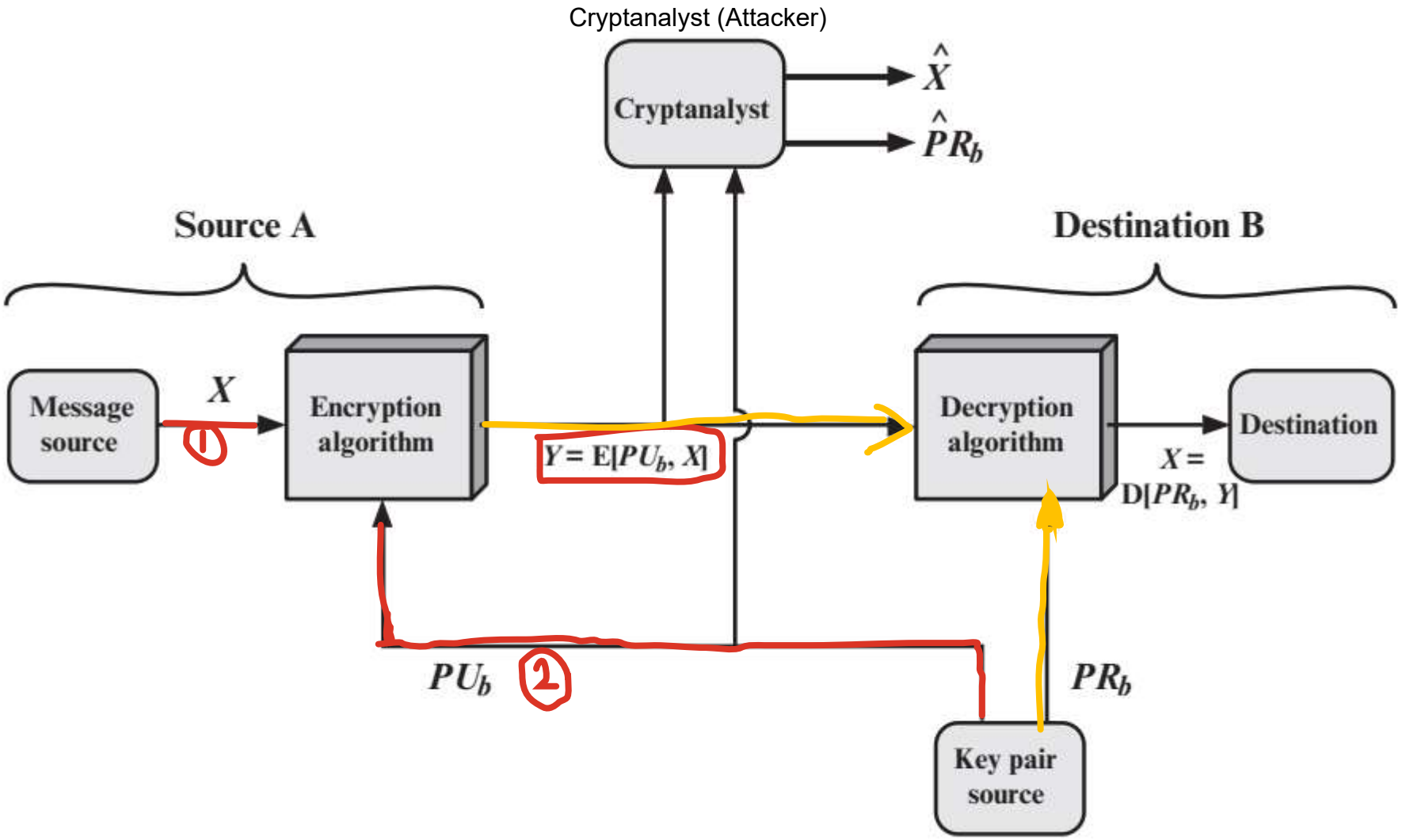


Figure 9.2 Public-Key Cryptosystem: Confidentiality

Authentication

- A prepares a message to B and encrypts it using A's private key before transmitting it.
- B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message.
- Therefore, the entire encrypted message serves as a **digital signature**
- In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.

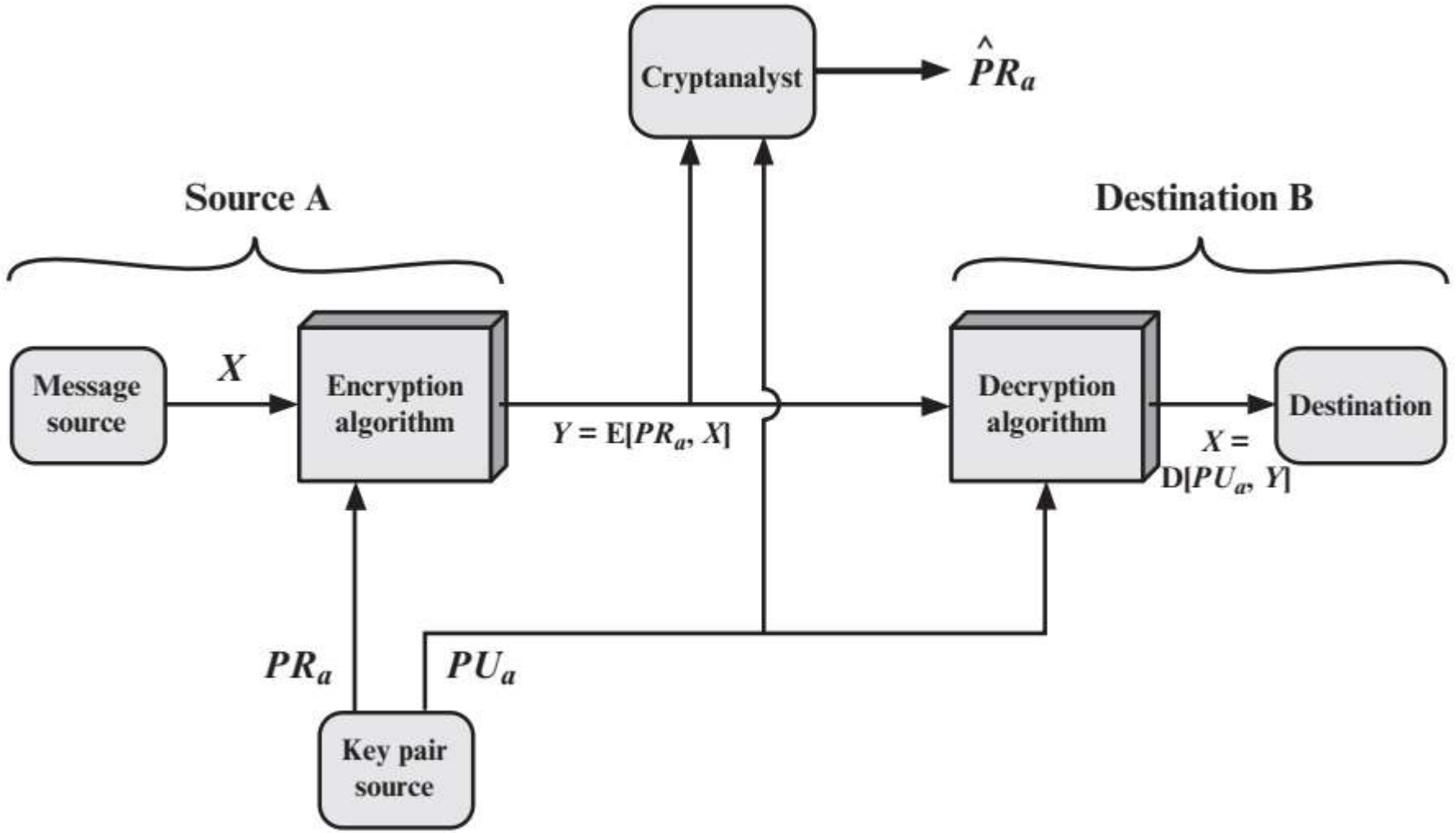


Figure 9.3 Public-Key Cryptosystem: Authentication

It is important to emphasize that the encryption process depicted in above fig does not provide confidentiality. the message being sent is safe from alteration but not from eavesdropping.

Authentication and Confidentiality

- By a double use of the public-key scheme:
  $Z = E(PUb, E(PRa, X))$
  $X = D(PUa, D(PRb, Z))$

- In this case, we begin as before by encrypting a message, using the sender's private key. This provides the digital signature.

- Next, we 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.

- The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

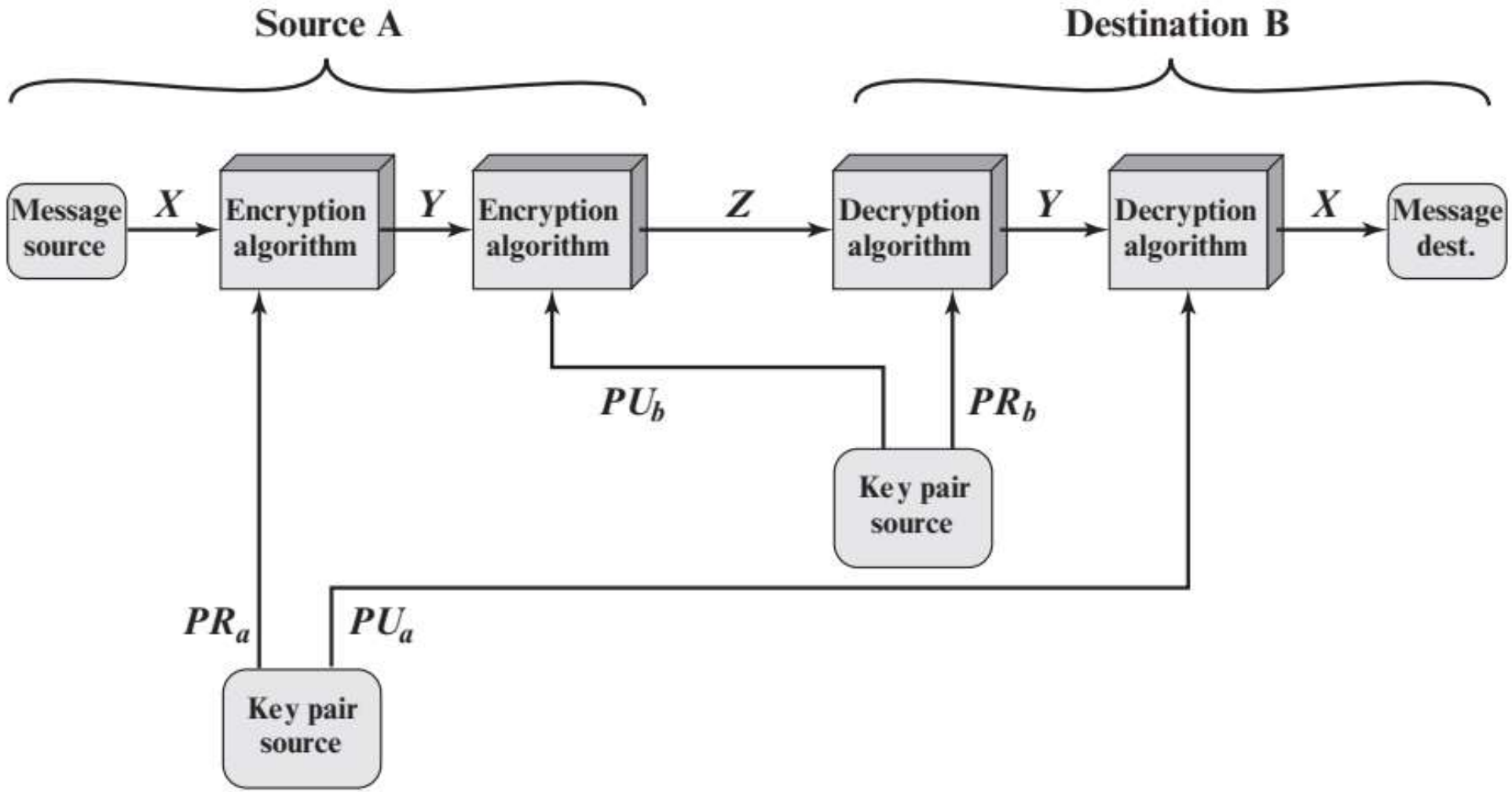


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Classification of Public-Key Cryptosystems

- we can classify the use of **public-key cryptosystems** into three categories
-
  - **Encryption/decryption:** The sender encrypts a message with the recipient's public key, and the recipient decrypts the message with the recipient's private key.
-
  - **Digital signature:** The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
-
  - **Key exchange:** Two sides cooperate to exchange a session key, which is a secret key for symmetric encryption generated for use for a particular session and valid for a short period of time.

Table 9.3 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

RSA Algorithm

- Was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978
- The Rivest-Shamir-Adleman (RSA) scheme is the most widely accepted and implemented general-purpose approach to public-key encryption.
- The **RSA** scheme is a cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n.
- A typical size for n is 1024 bits, or 309 decimal digits.
- RSA makes use of an expression with exponentials.
- Plaintext is encrypted in blocks, with each block having a binary value less than some number n.
- The block size must be less than or equal to $\log_2(n) + 1$
- Encryption and decryption are of the following form, for some plaintext block M and ciphertext 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.
- Thus, this is a public key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$.

- 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, and 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.

First condition: $M^{ed} \bmod n = M$

It holds if e and d are multiplicative inverses modulo $\Phi(n)$, where $\Phi(n)$ is the Euler totient function.

- Euler's totient function, also known as the phi function, counts the number of positive integers that are coprime to a given number.
- $\phi(8)=4$ because 1, 3, 5, and 7 are the only numbers less than 8 that are coprime to 8
- Co prime numbers are those numbers that have only one common factor, namely 1.

Determine $\phi(37)$ and $\phi(35)$.

Because 37 is prime, all of the positive integers from 1 through 36 are relatively prime to 37. Thus $\phi(37) = 36$.

To determine $\phi(35)$, we list all of the positive integers less than 35 that are relatively prime to it:

$$1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18$$
$$19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34$$

There are 24 numbers on the list, so $\phi(35) = 24$.

Table 2.6 Some Values of Euler's Totient Function $\phi(n)$

n	$\phi(n)$
1	1
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4

n	$\phi(n)$
11	10
12	4
13	12
14	6
15	8
16	8
17	16
18	6
19	18
20	8

n	$\phi(n)$
21	12
22	10
23	22
24	8
25	20
26	12
27	18
28	12
29	28
30	8

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p - 1) \times (q - 1)$$

Modular arithmetic

Properties of Congruences

Congruences have the following properties:

1. $a \equiv b \pmod{n}$ if $n \mid (a - b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$.
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

Modular arithmetic exhibits the following properties:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

Let $a, b \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$

If $a \equiv b \pmod{m}$, then $b \equiv a \pmod{m}$ — Symmetric Property

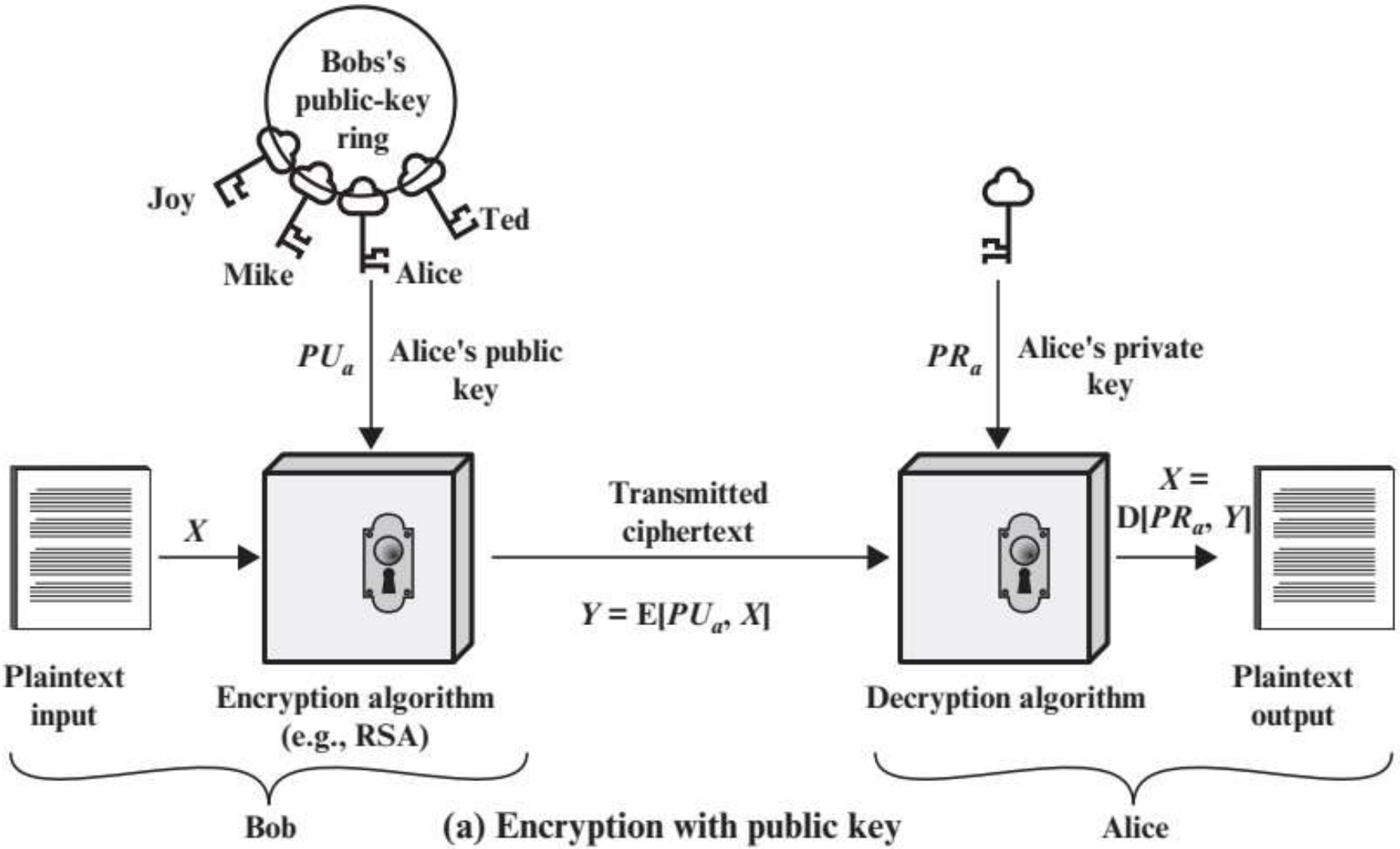
If $a \equiv b \pmod{m}$ and $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$ — Transitive Property

If $a \equiv b \pmod{m}$ then $ac = bc \pmod{mc}$ for $c > 0$ — Scalar Multiple

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then $a + c = (b + d) \pmod{m}$ — Addition Property

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then $ac = bd \pmod{m}$ — Multiplication Property

If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$ then $a - c = (b - d) \pmod{m}$ — Substraction Property



(a) Encryption with public key

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:	$M = C^d \bmod n$

Figure 9.5 The RSA Algorithm

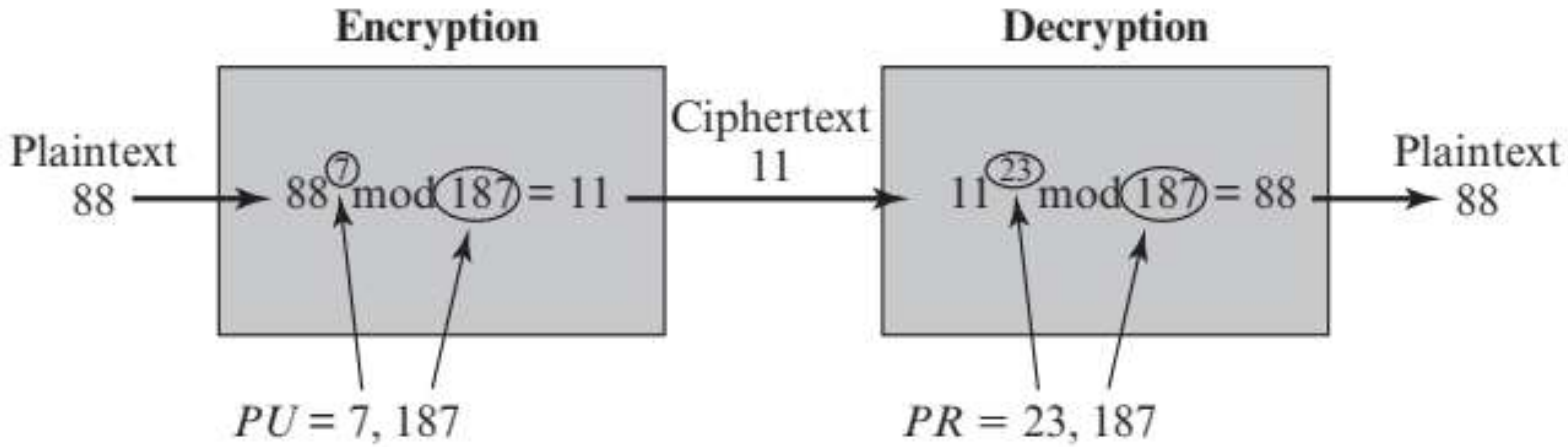


Figure 9.6 Example of RSA Algorithm

an example : showing the use of RSA to process multiple blocks of data. the plaintext is an alphanumeric string. Each plaintext symbol is assigned a unique code of two decimal digits (e.g., a = 00, A = 26).
A plaintext block consists of four decimal digits, or two alphanumeric characters.

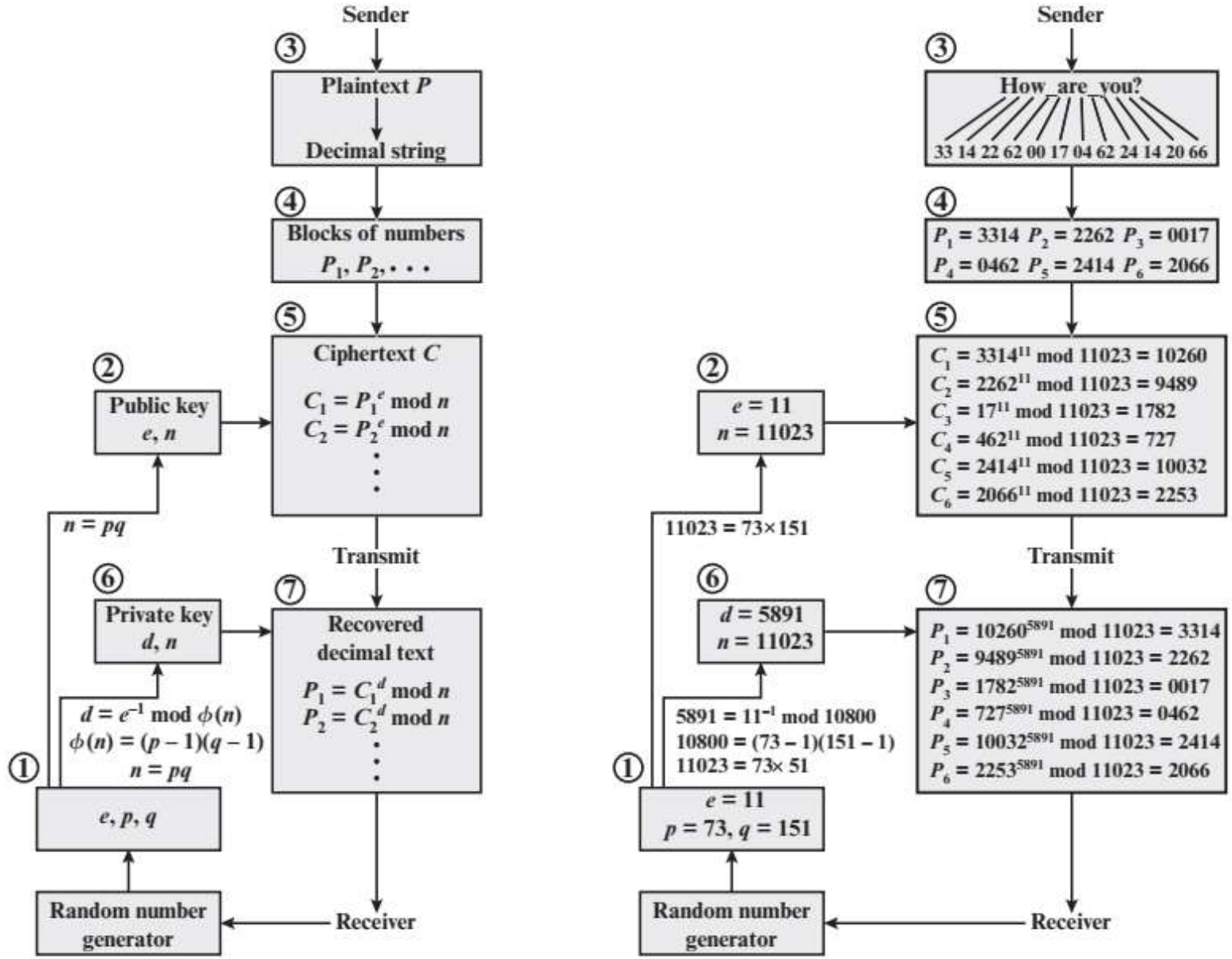


Figure 9.7 RSA Processing of Multiple Blocks

Diffie Hellman Key exchange

- The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages.
- It depends for its effectiveness on the difficulty of computing discrete logarithms.

- **Discrete logarithm**：
- The primitive root of a prime number is an integer that can be used to represent all numbers between 1 and the prime number minus one.

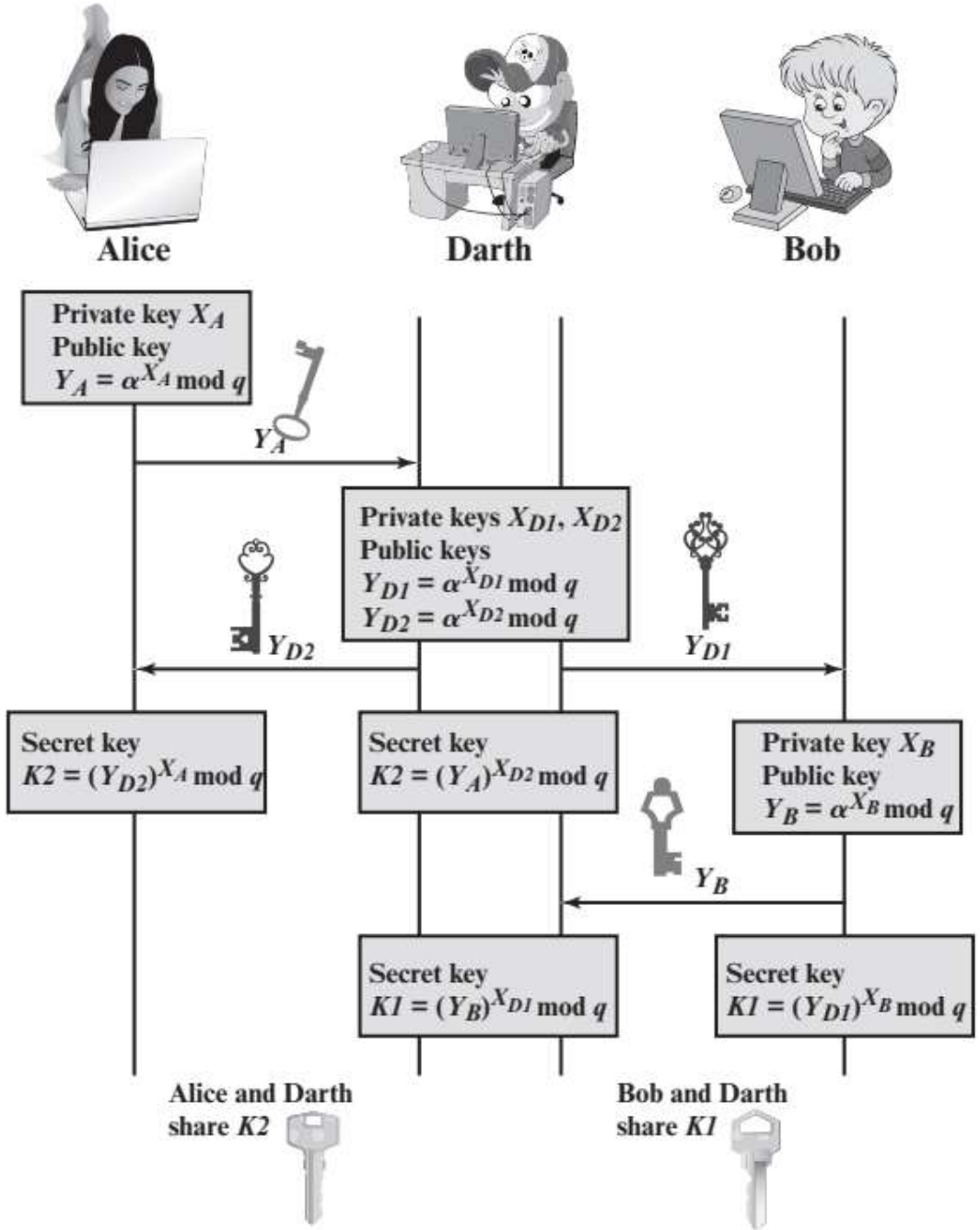


Figure 10.2 Man-in-the-Middle Attack

The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants.

This vulnerability can be overcome with the use of digital signatures and public-key certificates;