Asymmetric cryptography

CSS 325

Asymmetric Encryption Terms

- Asymmetric Keys

Two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

- Public Key Certificate

A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a subscriber to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key.

Asymmetric Encryption Terms

- Public Key (Asymmetric) Cryptographic Algorithm

A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that deriving the private key from the public key is computationally infeasible.

- Public Key Infrastructure (PKI)

A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.

Why Asymmetric

- Key distribution
  - ❖Two communicants already share a key
  - ❖The use of a key distribution center

The second requirement negated the very essence of cryptography:

The ability to maintain total secrecy over your own communication

Why Asymmetric Cont.

- Digital signatures
  - ❖The use of cryptography was to become widespread, not just in military situations but for commercial and private purposes.
  - ❖electronic messages and documents would need the equivalent of signatures used in paper documents
  - ❖Finding a method that would bring satisfaction to all parties, that a digital message had been sent by a particular person

Diffie and Hellman achieved an astounding breakthrough in 1976 that addressed both problems

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.
  - ❖ Either of the two related keys can be used for encryption, with the other used for decryption.

Public-key encryption scheme 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.
- 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.

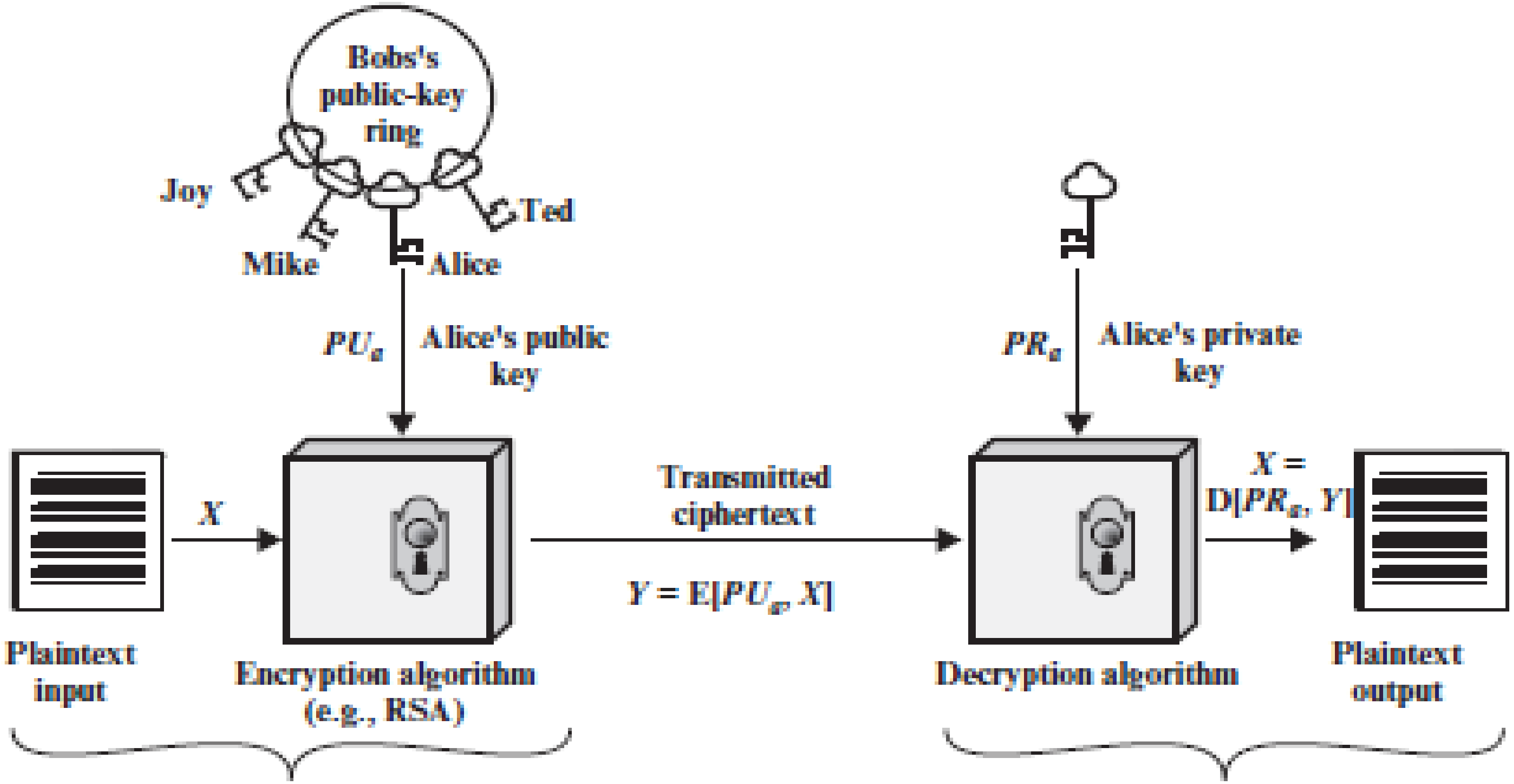
Public-key encryption scheme ingredients Cont.

- Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps

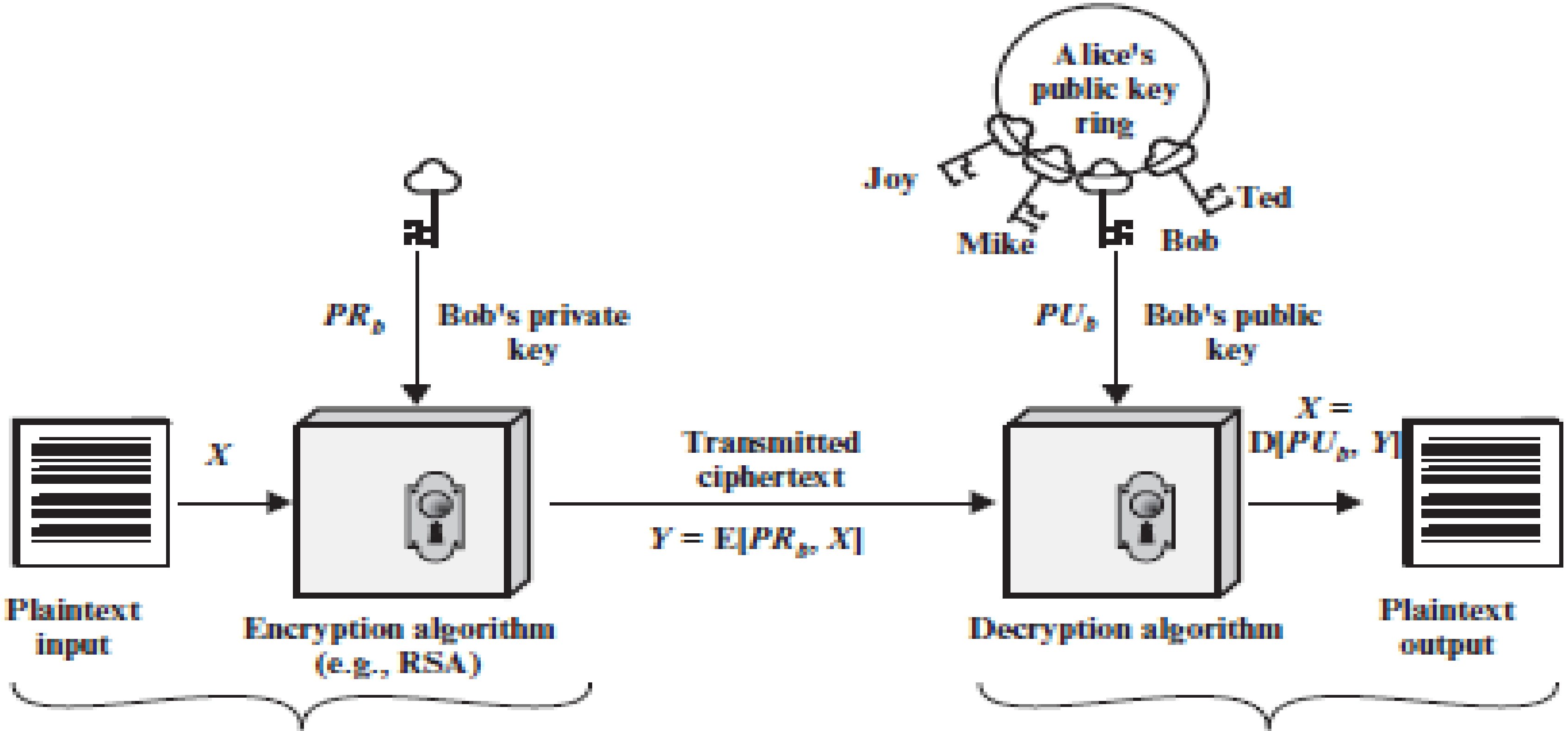
- Each user generates a pair of keys to be used for the encryption and decryption of messages.
- Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private
- If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice’s public key
- When Alice receives the message, she decrypts it using her private key

Encryption with public key



(a) Encryption with public key

Encryption with private key



(b) Encryption with private key

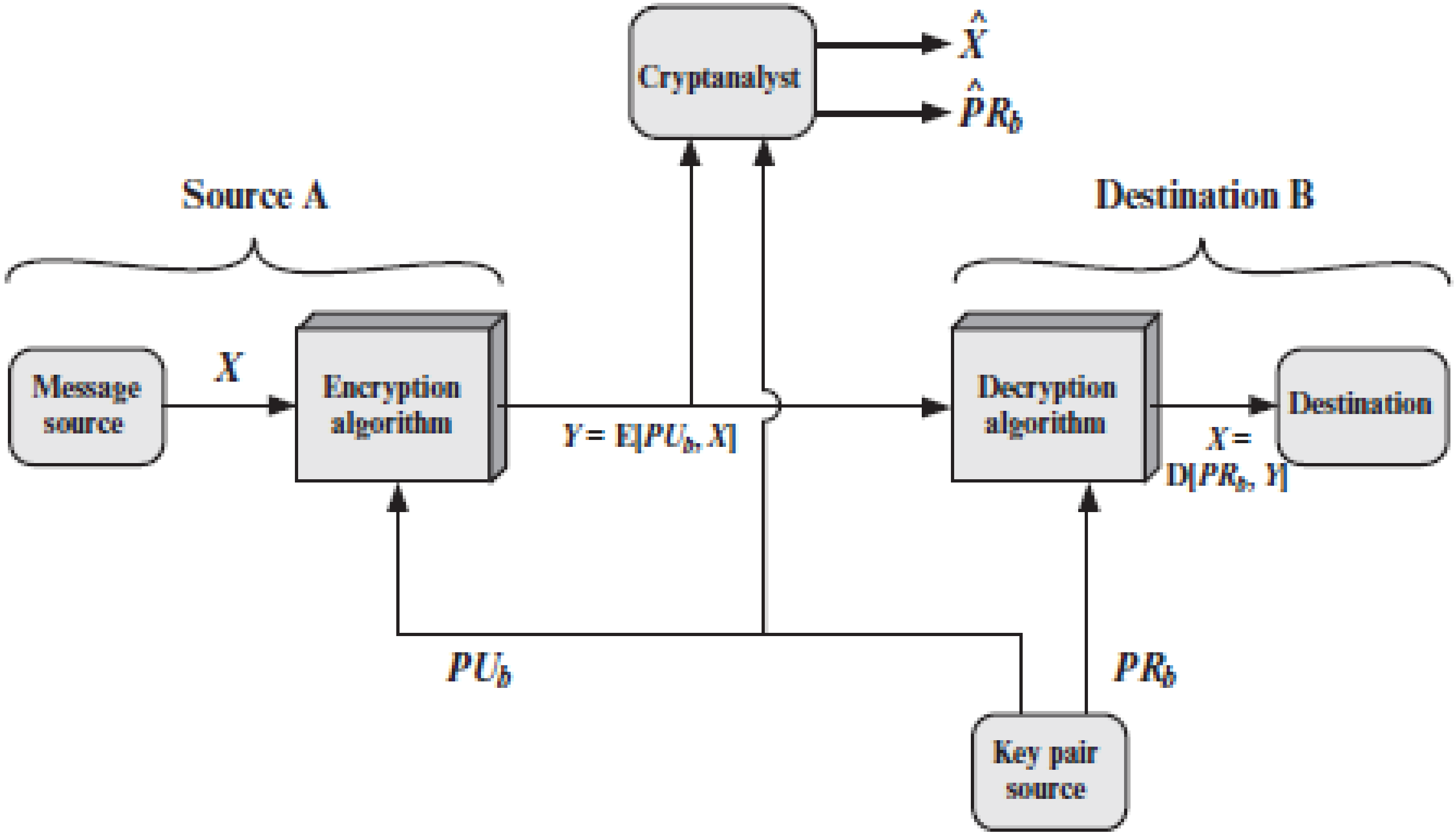
Conventional and Public-Key Encryption

Conventional Encryption	Public-Key Encryption
Needed to Work:	
The same algorithm with the same key is used for encryption and decryption.	One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption.
The sender and receiver must share the algorithm and the key	The sender and receiver must each have one of the matched pair of keys (not the same one).

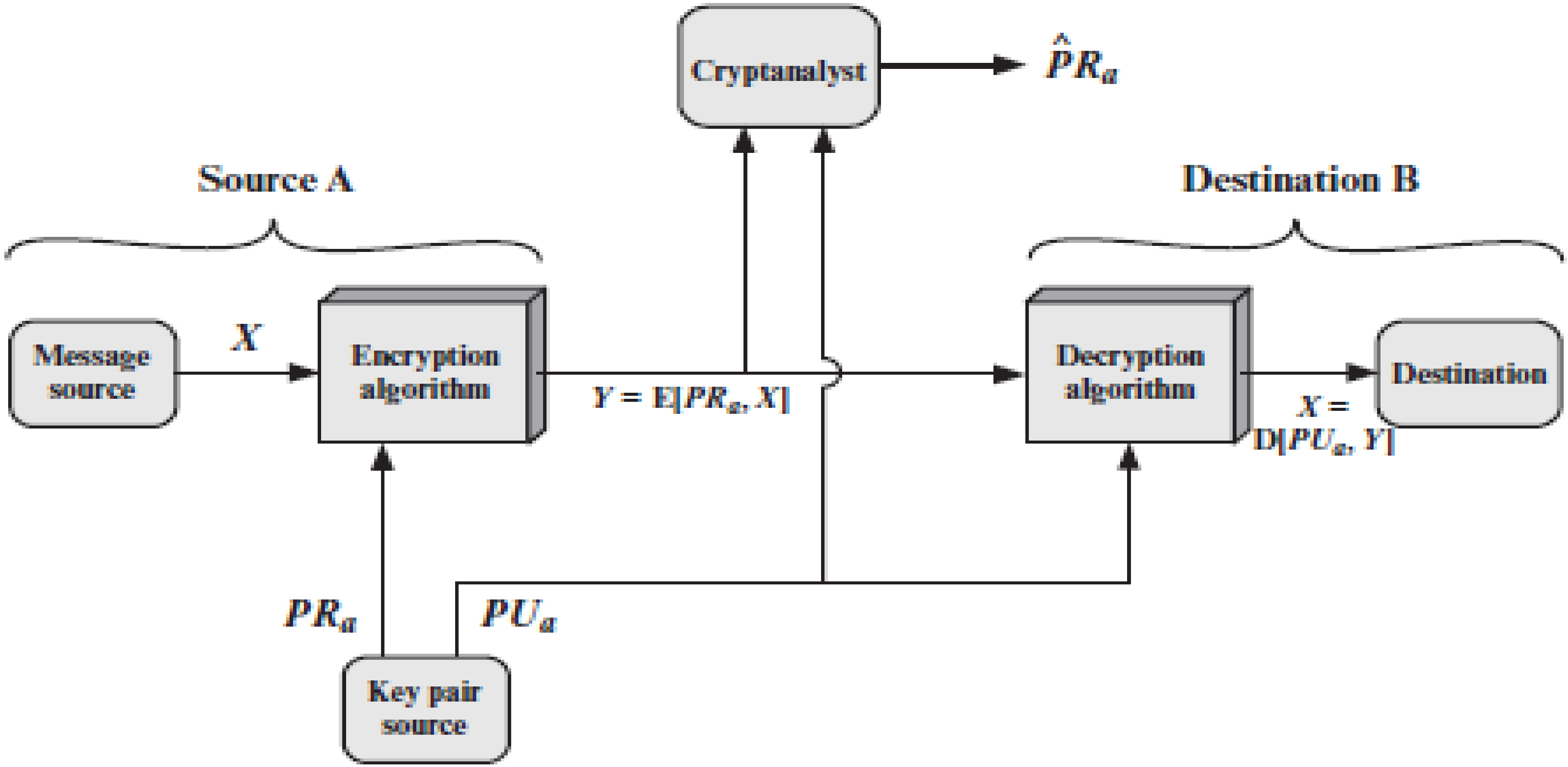
Conventional and Public-Key Encryption

Needed for Security:	
The key must be kept secret.	One of the two keys must be kept secret
It must be impossible or at least impractical to decipher a message if the key is kept secret.	It must be impossible or at least impractical to decipher a message if one of the keys is kept secret.
Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key

Public-Key Cryptosystem: Confidentiality



Public-Key Cryptosystem: Authentication

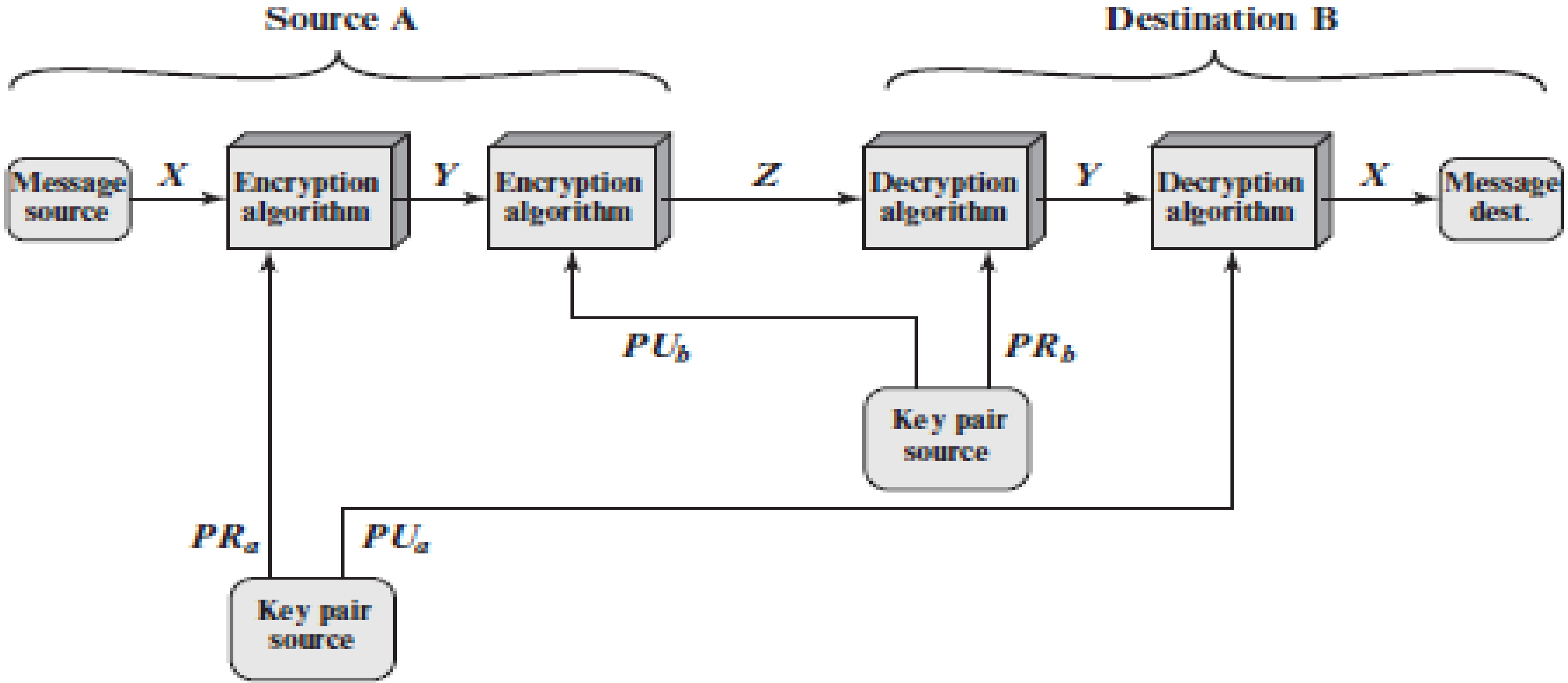


Authentication function and confidentiality

$$Z = \mathrm{E}(PU_b, \mathrm{E}(PR_a, X))$$

$$X = \mathrm{D}(PU_a, \mathrm{D}(PR_b, Z))$$

Cryptosystem: Authentication and Secrecy



Public-key cryptosystems Category

- **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 transaction (or session) and valid for a short period of time

Requirements for Public-Key Cryptography

- It is computationally easy for a party B to generate a key pair (public key PU_b, private key PR_b)
- It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:

$$C = \mathrm{E}(PU_b, M)$$

- It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

Requirements for Public-Key Cryptography Cont.

- It is computationally infeasible for an adversary, knowing the public key, PU_b, to determine the private key, PR_b
- It is computationally infeasible for an adversary, knowing the public key, PU_b, and a ciphertext, C, to recover the original message, M.
- The two keys can be applied in either order:

$$M = \mathrm{D}[PU_b, \mathrm{E}(PR_b, M)] = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

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 Cryptanalysis

- As with symmetric encryption, a public-key encryption scheme is vulnerable to a brute-force attack.

  Solution: Use large keys

- Way to compute the private key given the public key

  To date, it has not been mathematically proven that this form of attack is infeasible

- Probable-message attack

RSA

- $C = M^{e} \bmod n$
- $M = C^{d} \bmod n$

Example RSA

- Select two prime numbers, $p = 17$ and $q = 11$.
- Calculate $n = pq = 17 * 11 = 187$.
- Calculate $\phi(n) = (p - 1)(q - 1) = 16 * 10 = 160$.
- Select e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$; we choose $e = 7$
- Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$. The correct value is $d = 23$, because $23 * 7 = 161 = (1 * 160) + 1$; d can be calculated using the extended Euclid's algorithm
- The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$

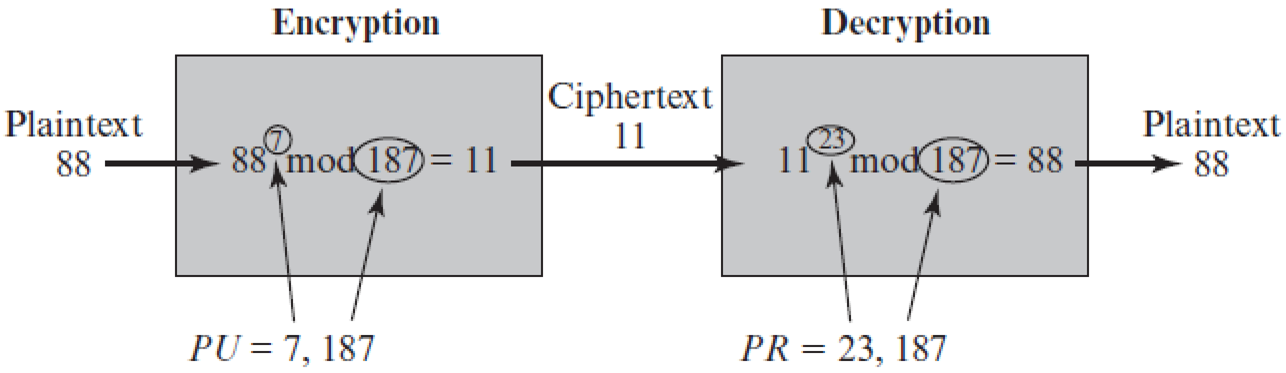
Example RSA Cont.

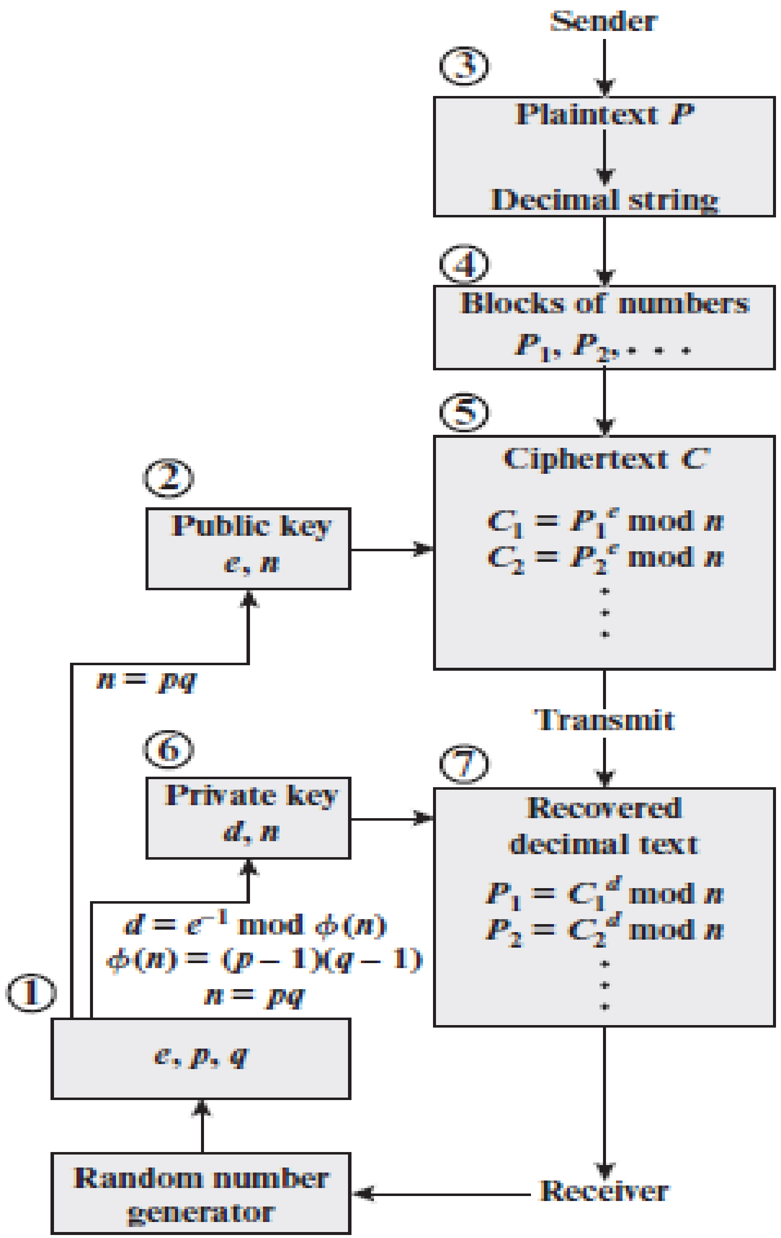
- $88^7 \bmod 18^7 = [(88^4 \bmod 187) * (88^2 \bmod 187) * (88^1 \bmod 187)] \bmod 187$
- $88^1 \bmod 187 = 88$
- $88^2 \bmod 187 = 7744 \bmod 187 = 77$
- $88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$
- $88^7 \bmod 187 = (88 * 77 * 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

Example RSA Cont.

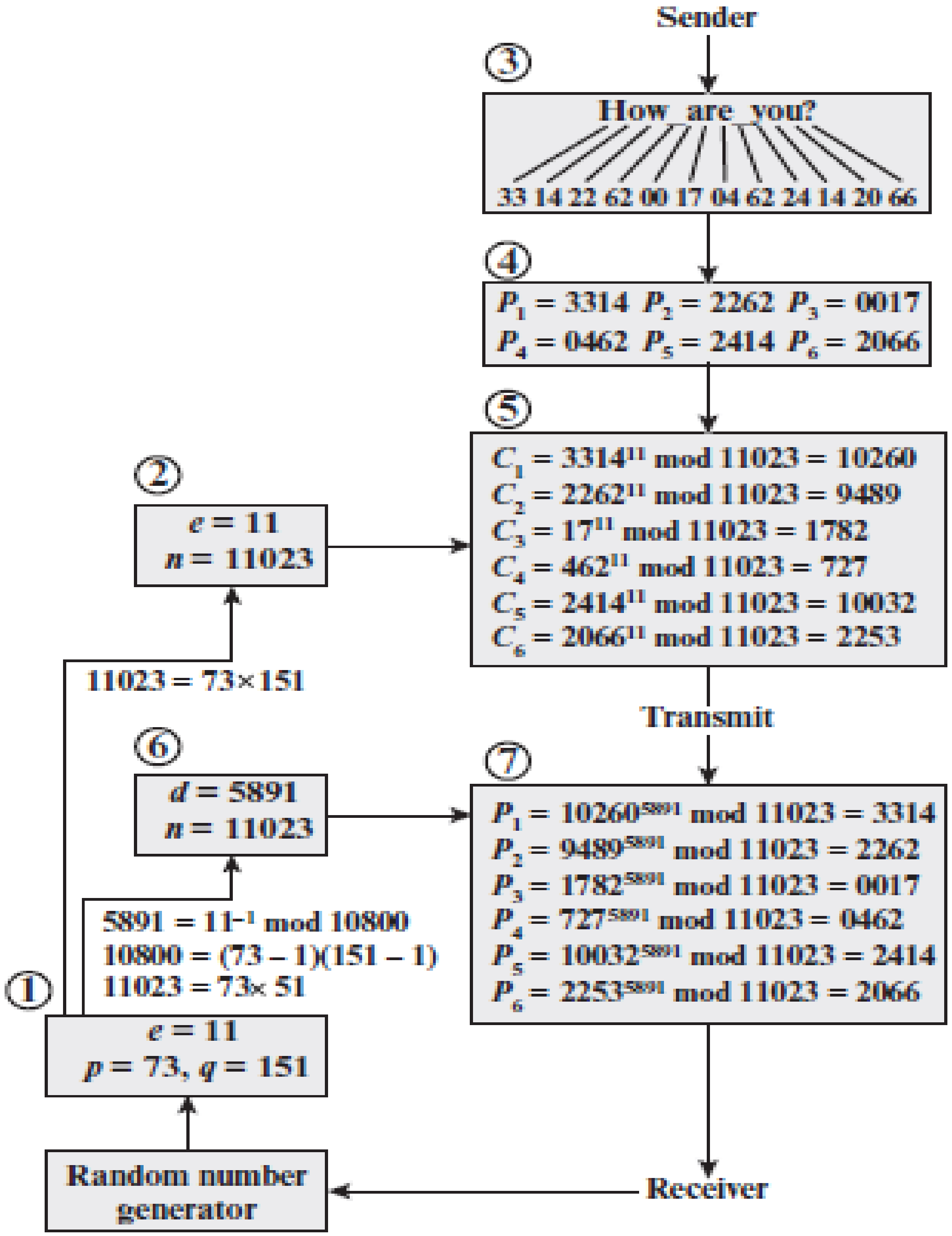
- For decryption, we calculate $M = 11^{23} \bmod 187$
- $11^{23} \bmod 187 = [(11^{1} \bmod 187) * (11^{2} \bmod 187) * (11^{4} \bmod 187) * (11^{8} \bmod 187) * (11^{8} \bmod 187)] \bmod 187$
- $11^{1} \bmod 187 = 11$
- $11^{2} \bmod 187 = 121$
- $11^{4} \bmod 187 = 14{,}641 \bmod 187 = 55$
- $11^{8}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

Example RSA Cont.





Sender
3
Plaintext P
Decimal string
4
Blocks of numbers
P1, P2, . . .
5
Ciphertext C
C1 = P1^e mod n
C2 = P2^e mod n
2
Public key
e, n
n = pq
Transmit
6
Private key
d, n
7
Recovered decimal text
P1 = C1^d mod n
P2 = C2^d mod n
d = e^-1 mod φ(n)
φ(n) = (p – 1)(q – 1)
n = pq
1
e, p, q
Random number generator
Receiver



Sender
3
How are you?
33 14 22 62 00 17 04 62 24 14 20 66
4
P1 = 3314 P2 = 2262 P3 = 0017
P4 = 0462 P5 = 2414 P6 = 2066
5
C1 = 3314^11 mod 11023 = 10260
C2 = 2262^11 mod 11023 = 9489
C3 = 17^11 mod 11023 = 1782
C4 = 462^11 mod 11023 = 727
C5 = 2414^11 mod 11023 = 10032
C6 = 2066^11 mod 11023 = 2253
2
e = 11
n = 11023
11023 = 73×151
Transmit
6
d = 5891
n = 11023
7
P1 = 10260^5891 mod 11023 = 3314
P2 = 9489^5891 mod 11023 = 2262
P3 = 1782^5891 mod 11023 = 0017
P4 = 727^5891 mod 11023 = 0462
P5 = 10032^5891 mod 11023 = 2414
P6 = 2253^5891 mod 11023 = 2066
5891 = 11^-1 mod 10800
10800 = (73 – 1)(151 – 1)
11023 = 73× 51
1
e = 11
p = 73, q = 151
Random number generator
Receiver

Cryptographic hash functions

- A hash function **H** accepts a variable-length block of data M as input and produces a fixed-size hash value **h = H(M).**
- In general terms, the principal object of a hash function is data integrity
- The kind of hash function needed for security applications is referred to as a **Cryptographic hash function**

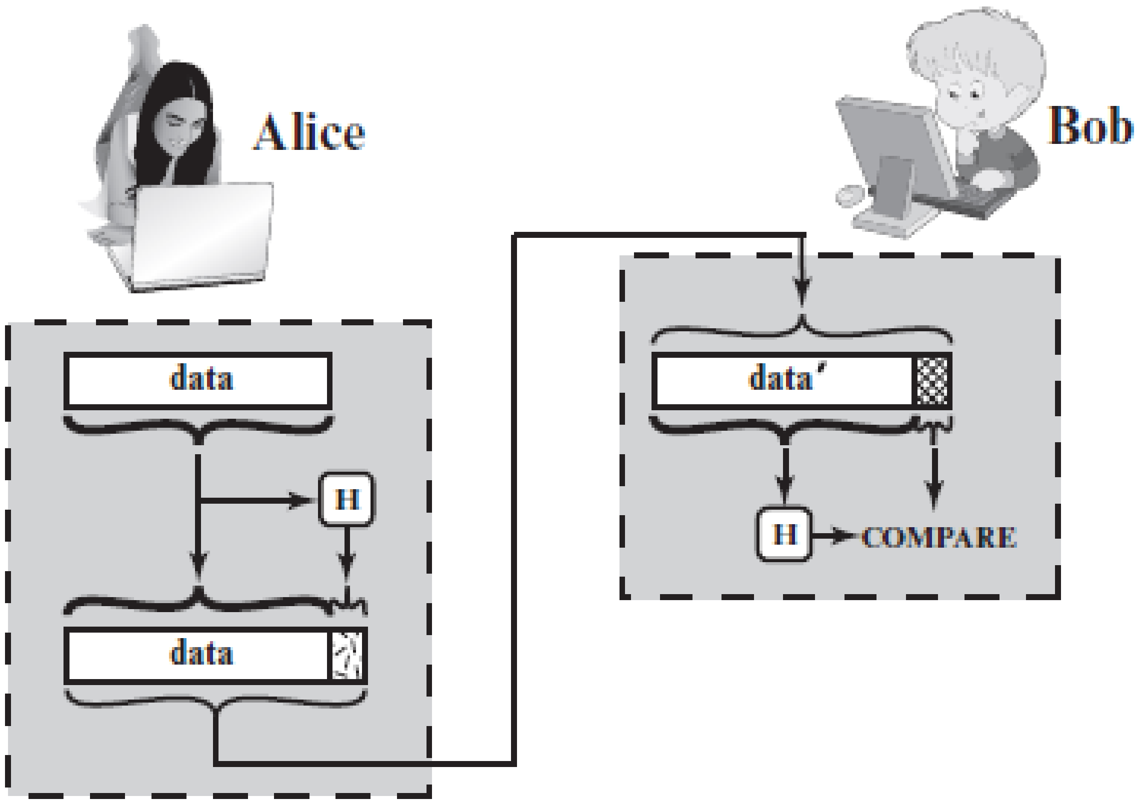
Applications of cryptographic hash functions

- Message Authentication
- Digital Signatures
- One-way password file
- Intrusion detection
- Virus detection

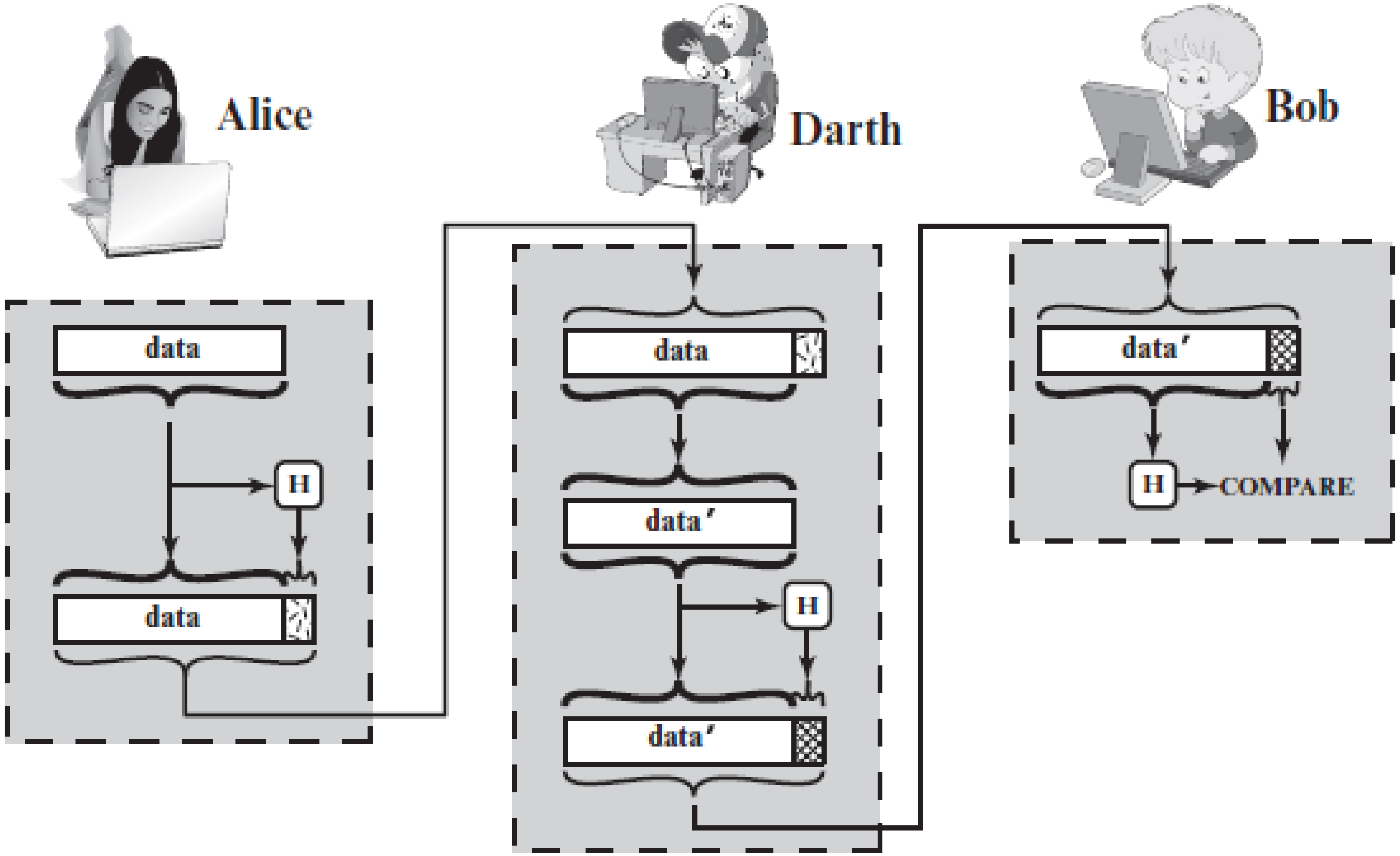
Message Authentication

- Message authentication is a mechanism or service used to verify the integrity of a message
- When a hash function is used to provide message authentication, the hash function value is often referred to as a **message digest**

Use of hash function to check data integrity

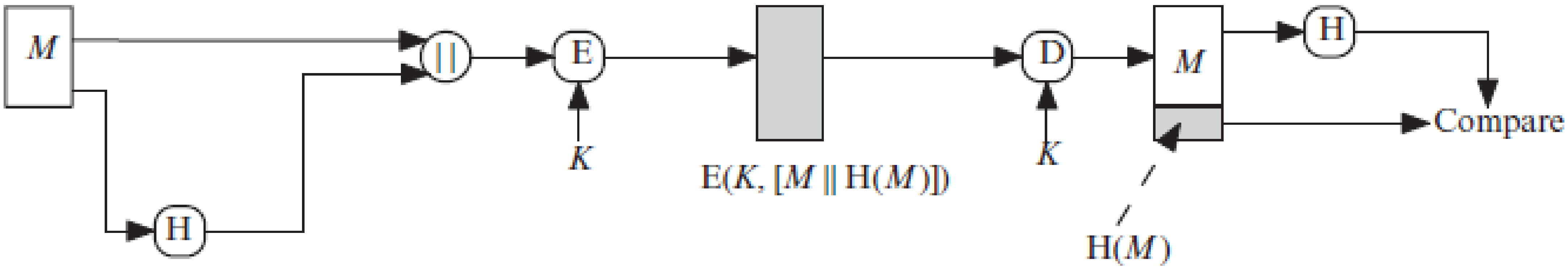


Attack Against Hash Function



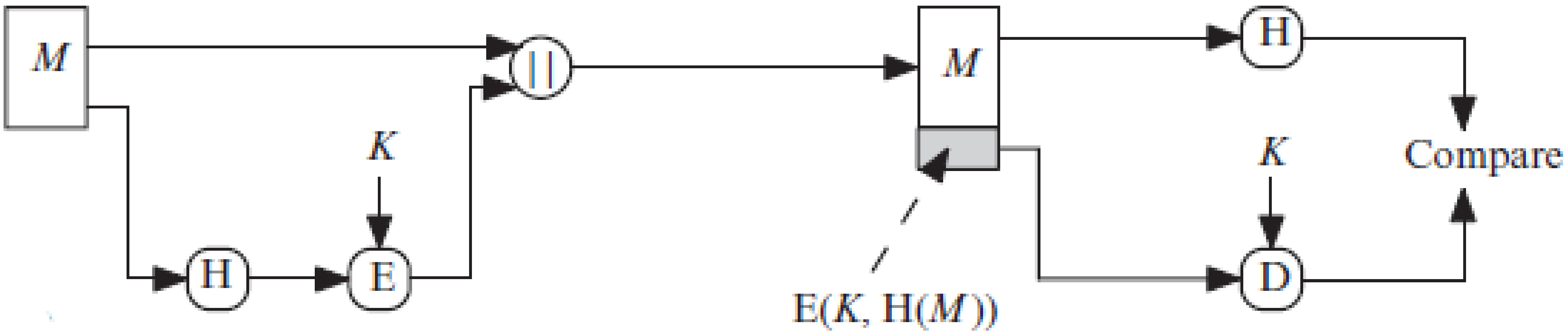
Ways in which a hash code can be used to provide message authentication

(a)



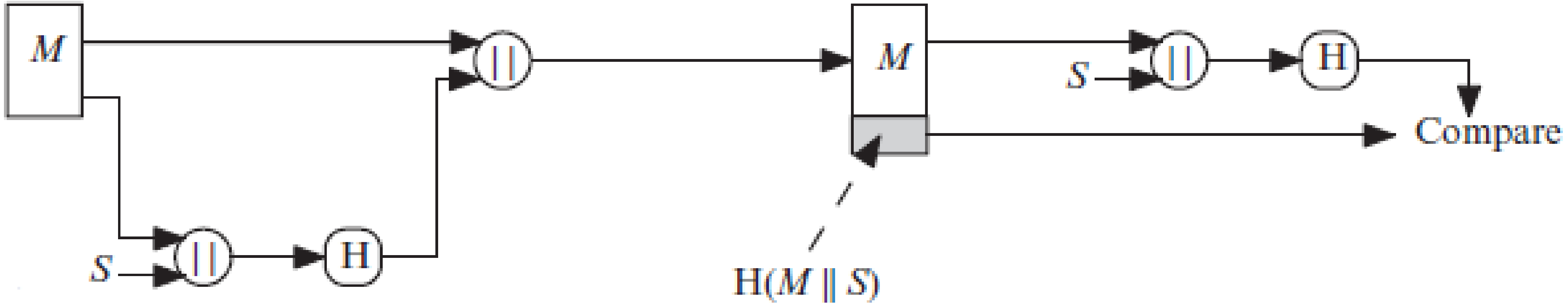
Ways in which a hash code can be used to provide message authentication

(b)



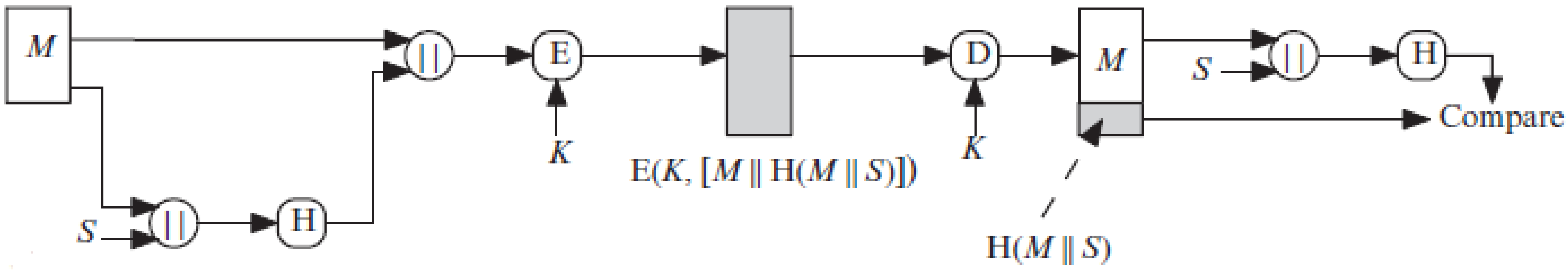
Ways in which a hash code can be used to provide message authentication

(c)



Ways in which a hash code can be used to provide message authentication

(d)



Digital Signatures

- Message Authentication Code (MAC)

A MAC function takes as input a secret key and a data block and produces a hash value, referred to as the MAC, which is associated with the protected message

- The operation of the **Digital Signature** is similar to that of the MAC
- In **Digital Signature** the hash value of a message is encrypted with a user's private key

Hash code providing a Digital Signature