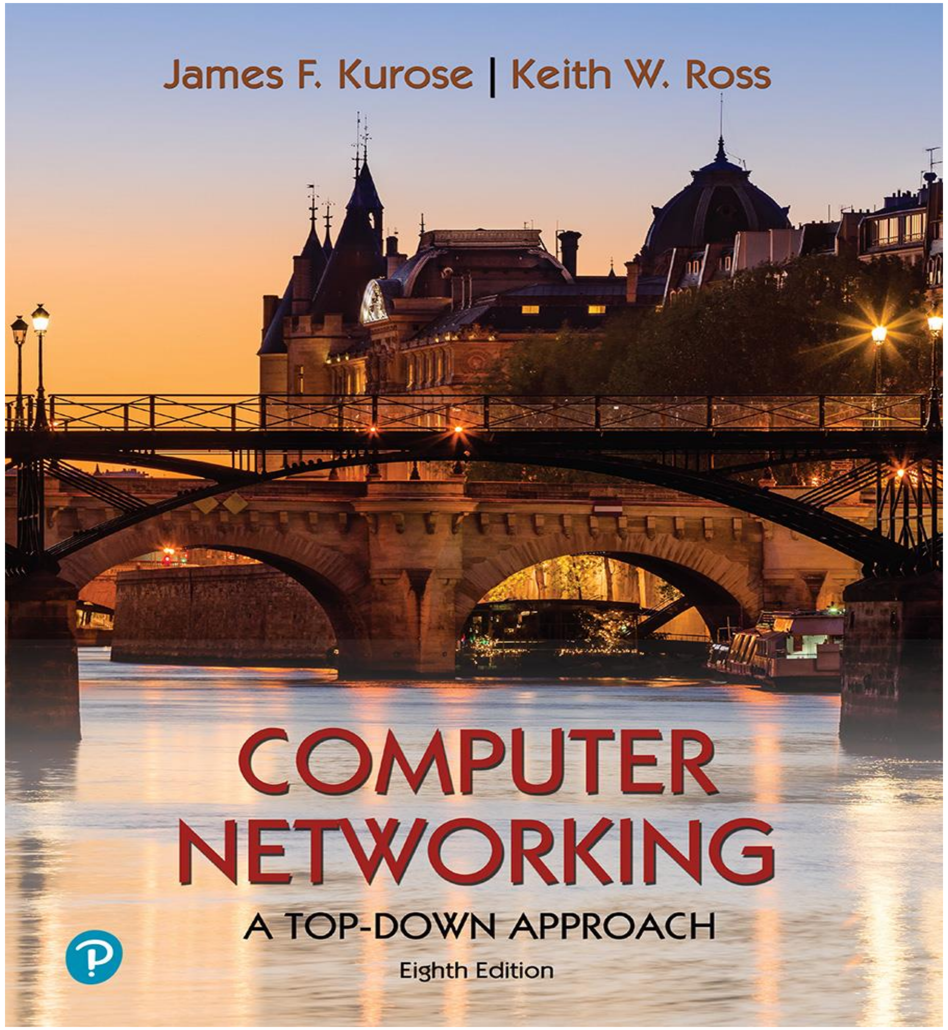
Network Security

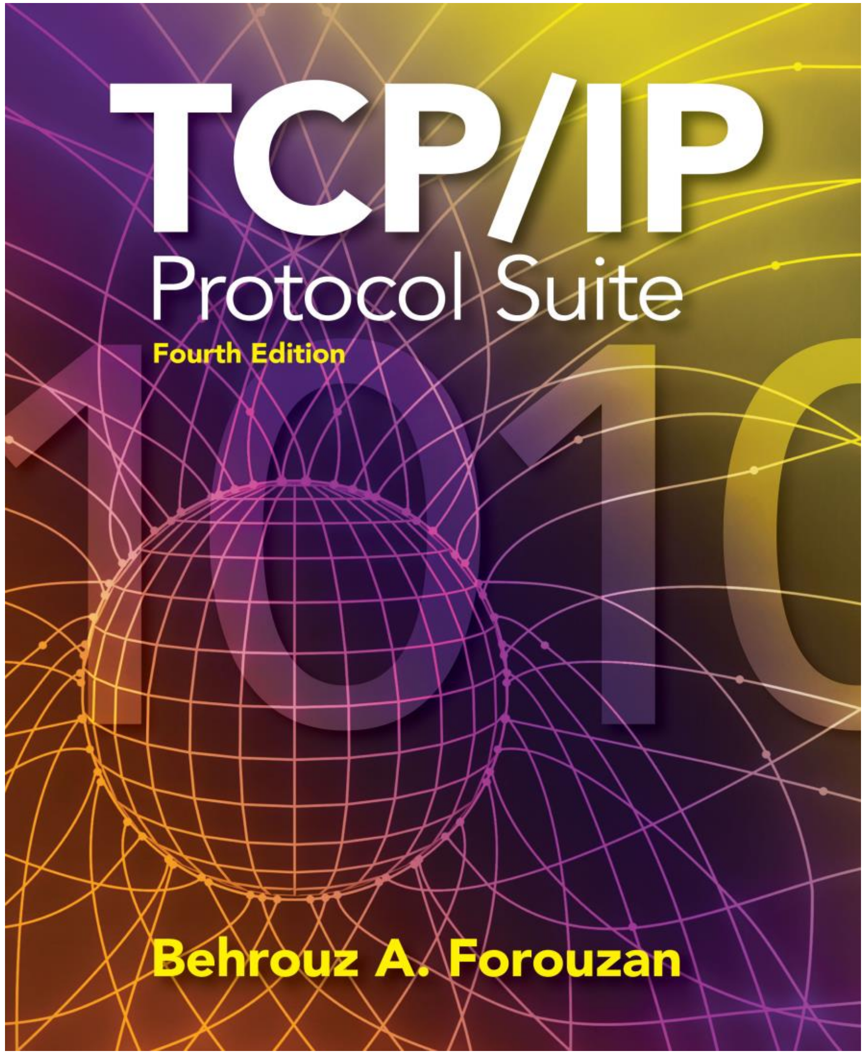
Anand Baswade

anand@iitbhilai.ac.in

Sources



Computer Networking: A Top-Down Approach



TCP/IP Protocol Suite

Security: overview

Chapter goals:

- understand principles of network security:
  - cryptography and its *many* uses beyond "confidentiality"
  - authentication
  - message integrity
- security in practice:
  - firewalls and intrusion detection systems
  - security in application, transport, network, link layers

Chapter 8 outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Operational security: firewalls and IDS

What is network security?

confidentiality: only sender, intended receiver should "understand" message contents

- sender encrypts message
- receiver decrypts message

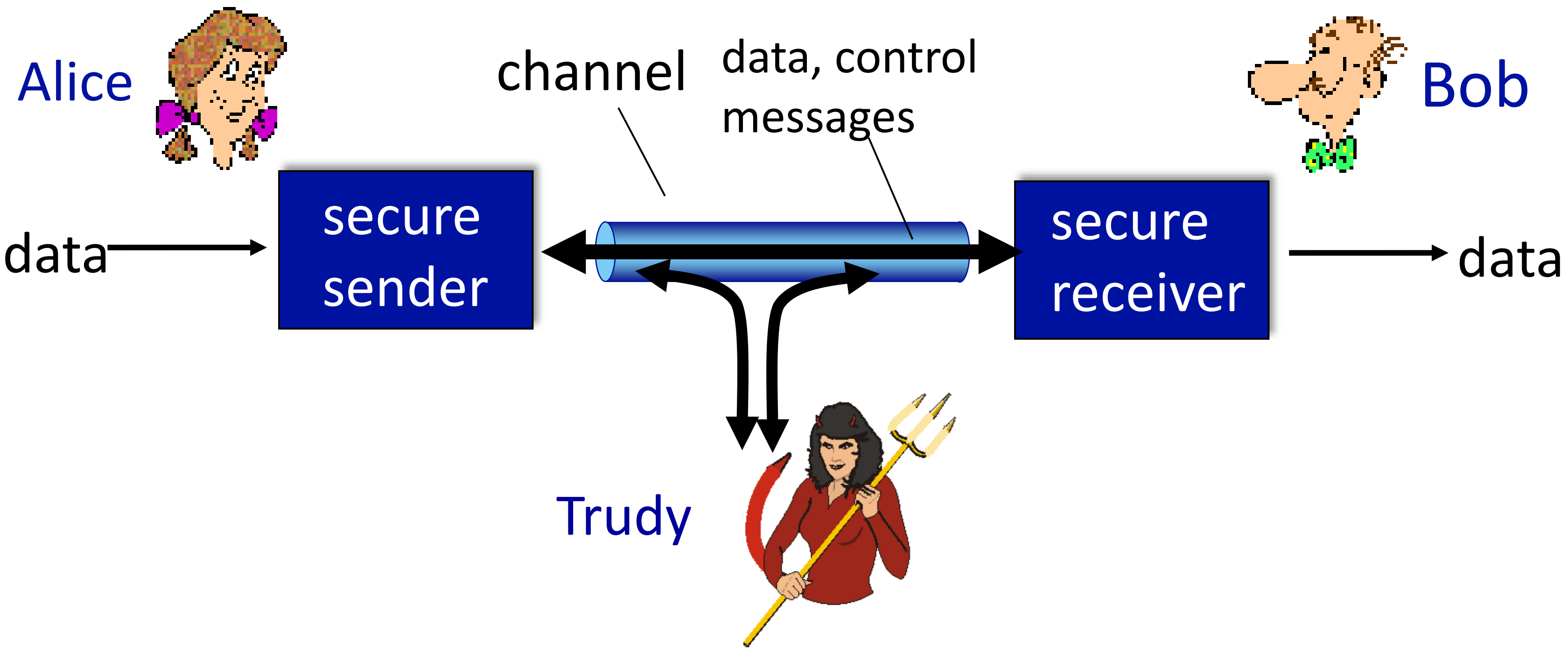
authentication: sender, receiver want to confirm identity of each other

message integrity: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection

access and availability: services must be accessible and available to users

Friends and enemies: Alice, Bob, Trudy

- well-known in network security world
- Bob, Alice want to communicate “securely”
- Trudy (intruder) may intercept, delete, add messages



Friends and enemies: Alice, Bob, Trudy

Who might Bob and Alice be?

- … well, *real-life* Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- BGP routers exchanging routing table updates
- other examples?

There are bad guys (and girls) out there!

Q: What can a "bad guy" do?

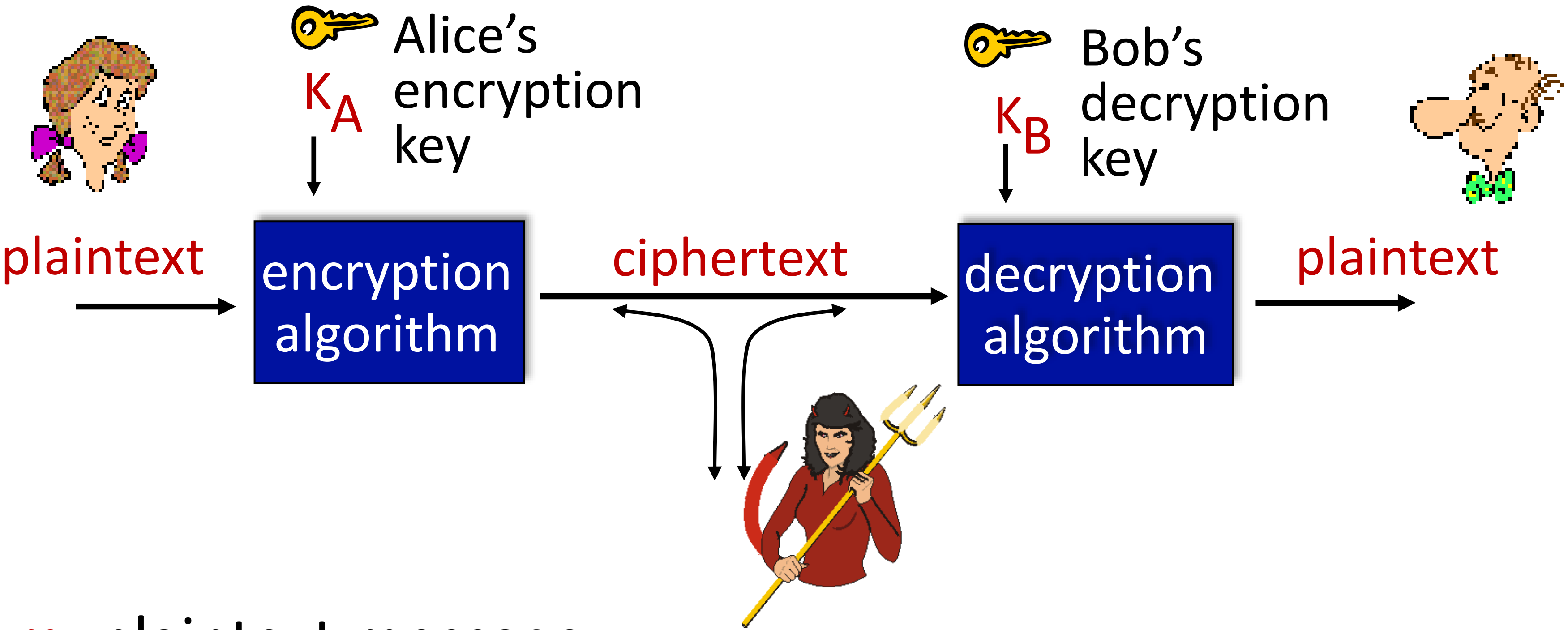
A: A lot!

- eavesdrop: intercept messages (secretly listen to a conversation)
- actively insert messages into connection
- impersonation: can fake (spoof) source address in packet (or any field in packet)
- hijacking: "take over" ongoing connection by removing sender or receiver, inserting himself in place
- denial of service: prevent service from being used by others (e.g., by overloading resources)

Chapter 8 outline

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Operational security: firewalls and IDS

The language of cryptography

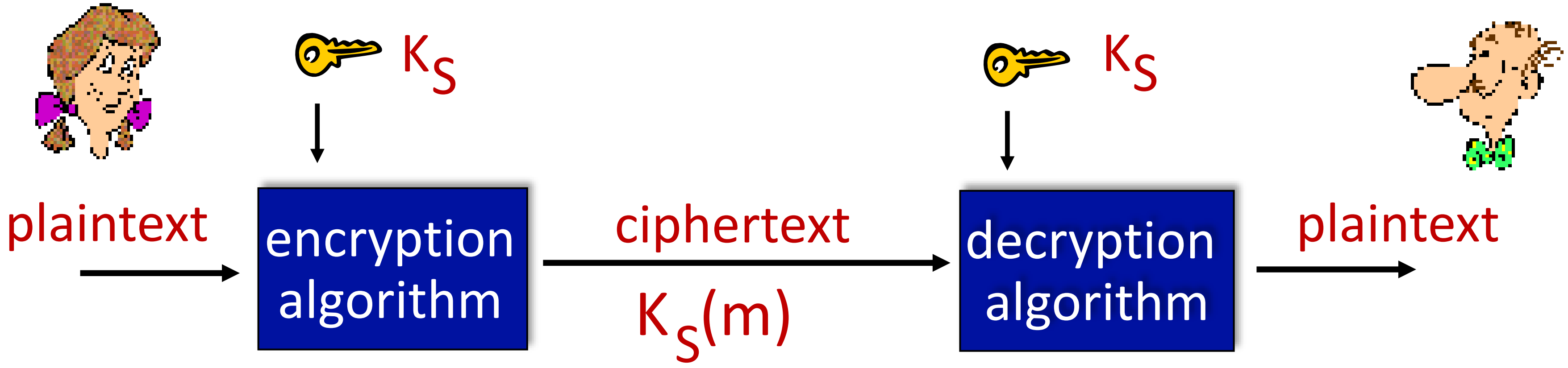


m: plaintext message

$K_A(m)$: ciphertext, encrypted with key K_A

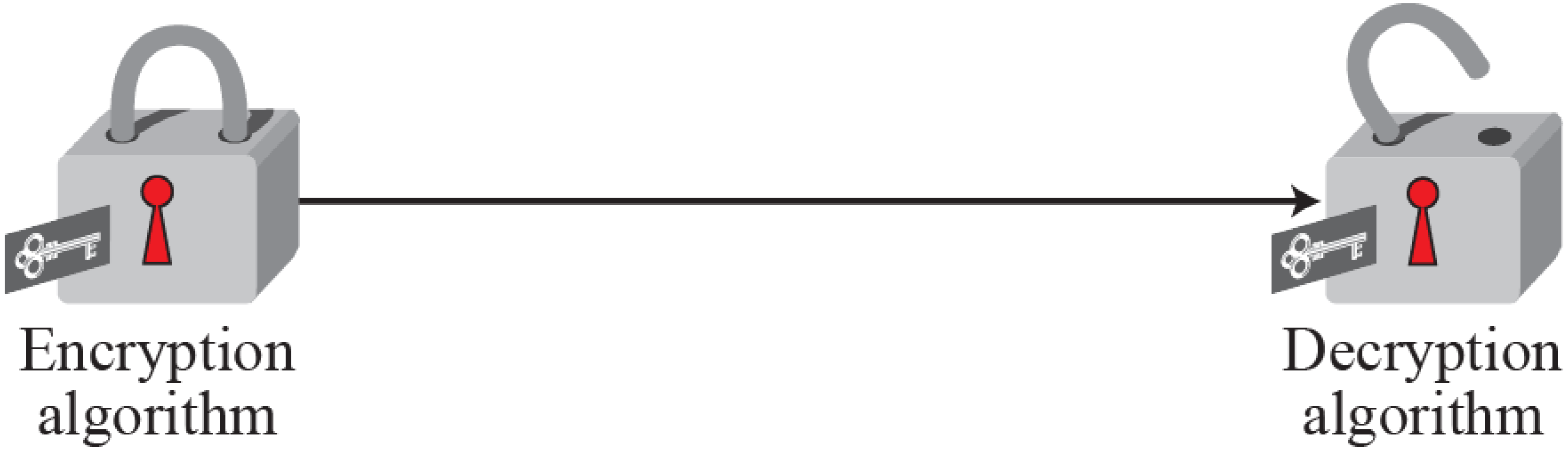
$m = K_B(K_A(m))$

Symmetric key cryptography

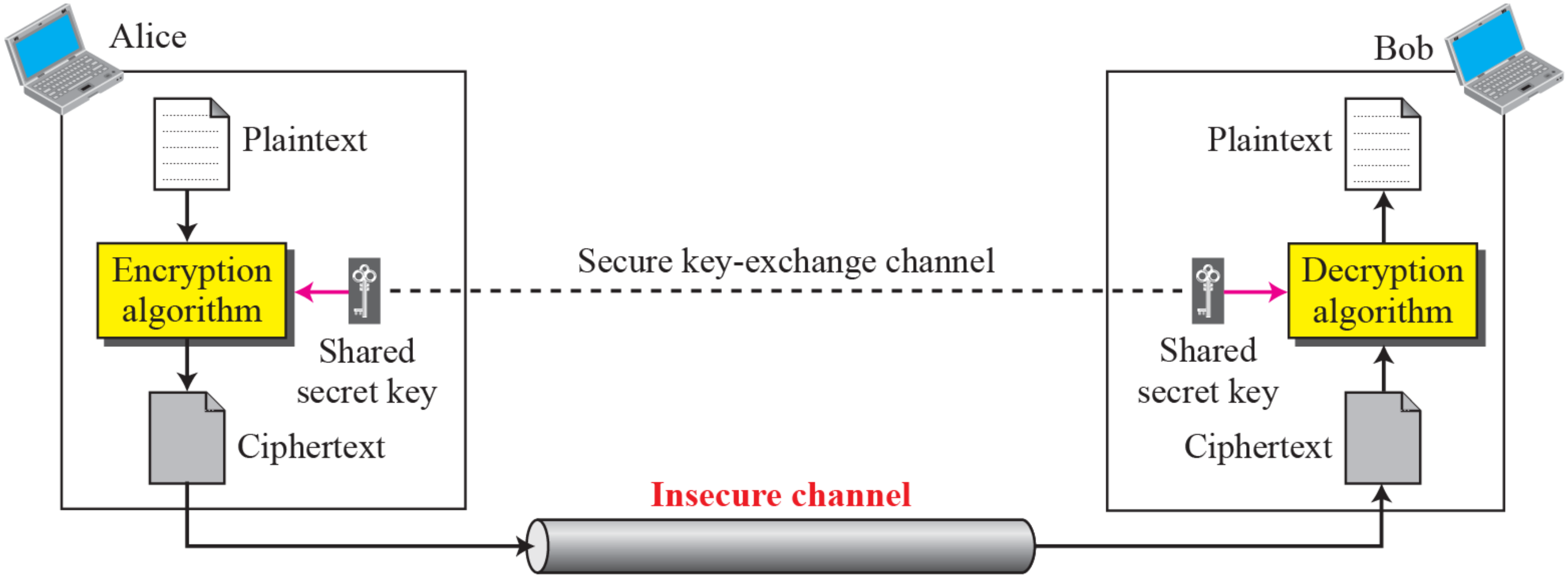


symmetric key crypto: Bob and Alice share same (symmetric) key: K

Symmetric-key: locking and unlocking with the same key



General idea of traditional cipher



Simple encryption scheme

substitution cipher: substituting one thing for another

- monoalphabetic cipher: substitute one letter for another

```
 plaintext:   abcdefghijklmnopqrstuvwxyz

ciphertext:   mnbvcxzasdfghjklpoiuytrewq
```

e.g.:
```
 Plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc
```

Encryption key: mapping from set of 26 letters to set of 26 letters

A more sophisticated encryption approach

- n substitution ciphers, $M_1,M_2,\ldots,M_n$
- cycling pattern:
  - e.g., n=4: M_1,M_3,M_4,M_3,M_2; M_1,M_3,M_4,M_3,M_2; ..
- for each new plaintext symbol, use subsequent substitution pattern in cyclic pattern
  - dog: d from M_1, o from M_3, g from M_4

Encryption key: n substitution ciphers, and cyclic pattern

- key need not be just n-bit pattern

Example

Use the additive cipher with key = 15 to encrypt the message “hello”.

Solution

We apply the encryption algorithm to the plaintext, character by character. The result is “WTAAD”. Note that the cipher is monoalphabetic because two instances of the same plaintext character (ls) are encrypted as the same character (A).

Plaintext: h $\rightarrow$ 07	Encryption: $(07 + 15) \bmod 26$	Ciphertext: 22 $\rightarrow$ W
Plaintext: e $\rightarrow$ 04	Encryption: $(04 + 15) \bmod 26$	Ciphertext: 19 $\rightarrow$ T
Plaintext: l $\rightarrow$ 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 $\rightarrow$ A
Plaintext: l $\rightarrow$ 11	Encryption: $(11 + 15) \bmod 26$	Ciphertext: 00 $\rightarrow$ A
Plaintext: o $\rightarrow$ 14	Encryption: $(14 + 15) \bmod 26$	Ciphertext: 03 $\rightarrow$ D

Example

Use the additive cipher with key = 15 to decrypt the message "WTAAD".

Solution

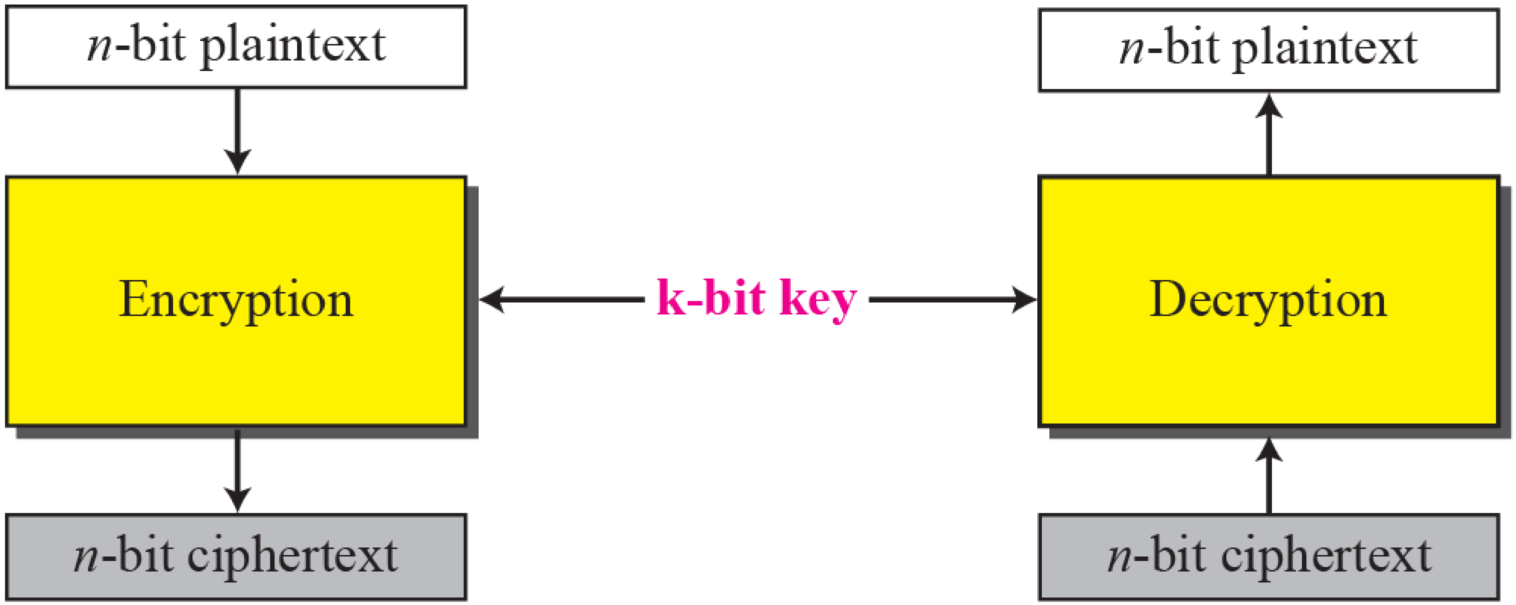
We apply the decryption algorithm to the plaintext character by character. The result is "hello". Note that the operation is in modulo 26, which means that we need to add 26 to a negative result (for example –15 becomes 11).

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

Modern Cipher

- The traditional symmetric-key ciphers that we have studied so far are character-oriented ciphers. With the advent of the computer, we need bit-oriented ciphers.
- This is because the information to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data.
- It is convenient to convert these types of data into a stream of bits, to encrypt the stream, and then to send the encrypted stream.
- A modern block cipher can be either a block cipher or a stream (bit by bit) cipher.

A modern block cipher



Symmetric key crypto: DES

DES: Data Encryption Standard

- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64-bit plaintext input
- block cipher with cipher block chaining
- how secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase decrypted (brute force) in less than a day
  - no known good analytic attack
- making DES more secure:
  - 3DES: encrypt 3 times with 3 different keys

AES: Advanced Encryption Standard

- symmetric-key NIST standard, replaced DES (Nov 2001)
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Asymmetric Key Cryptography (Public Key Crypto)

- In previous sections we discussed symmetric-key ciphers. In this chapter, we start the discussion of asymmetric-key ciphers.

- Symmetric-key and asymmetric-key ciphers will exist in parallel and continue to serve the community.

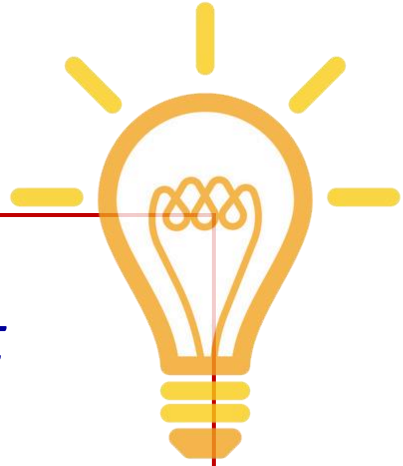
- We actually believe that they are complements of each other; the advantages of one can compensate for the disadvantages of the other.

Public Key Cryptography

symmetric key crypto:

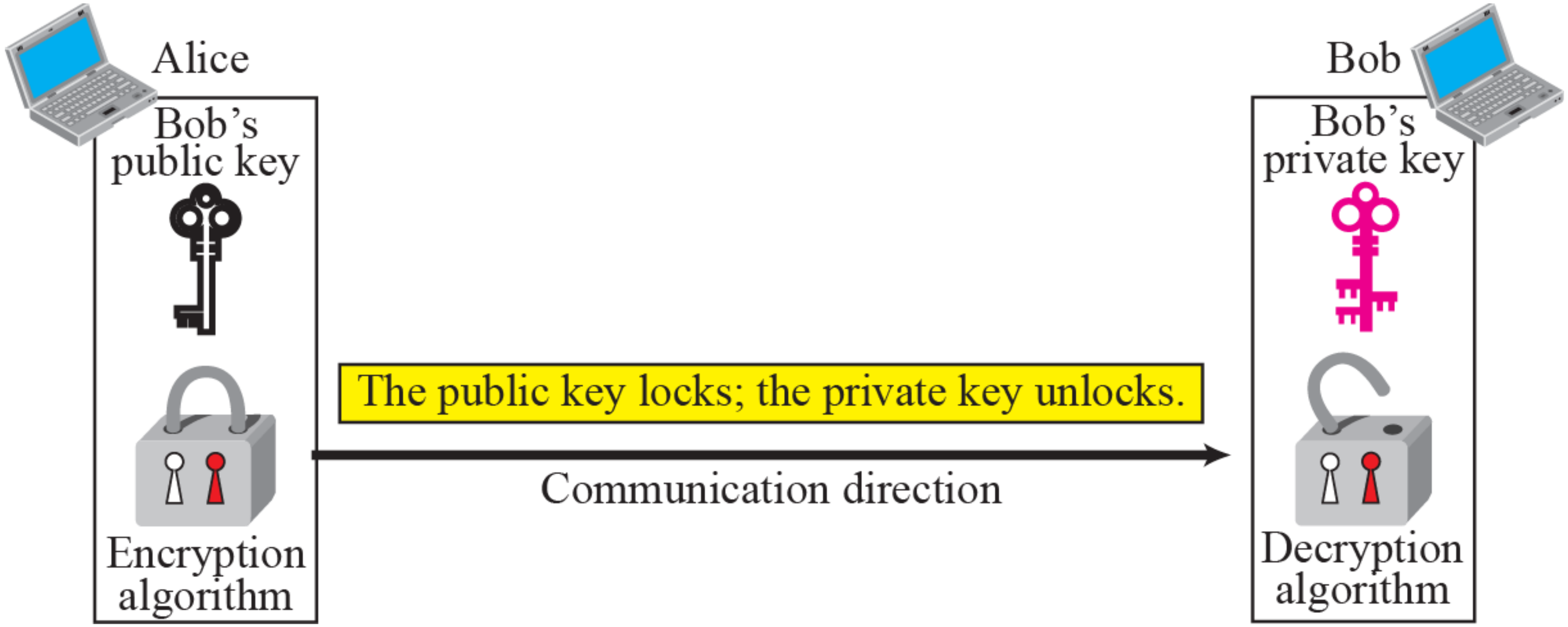
- requires sender, receiver know shared secret key
- Q: how to agree on key in first place (particularly if never “met”)?

public key crypto

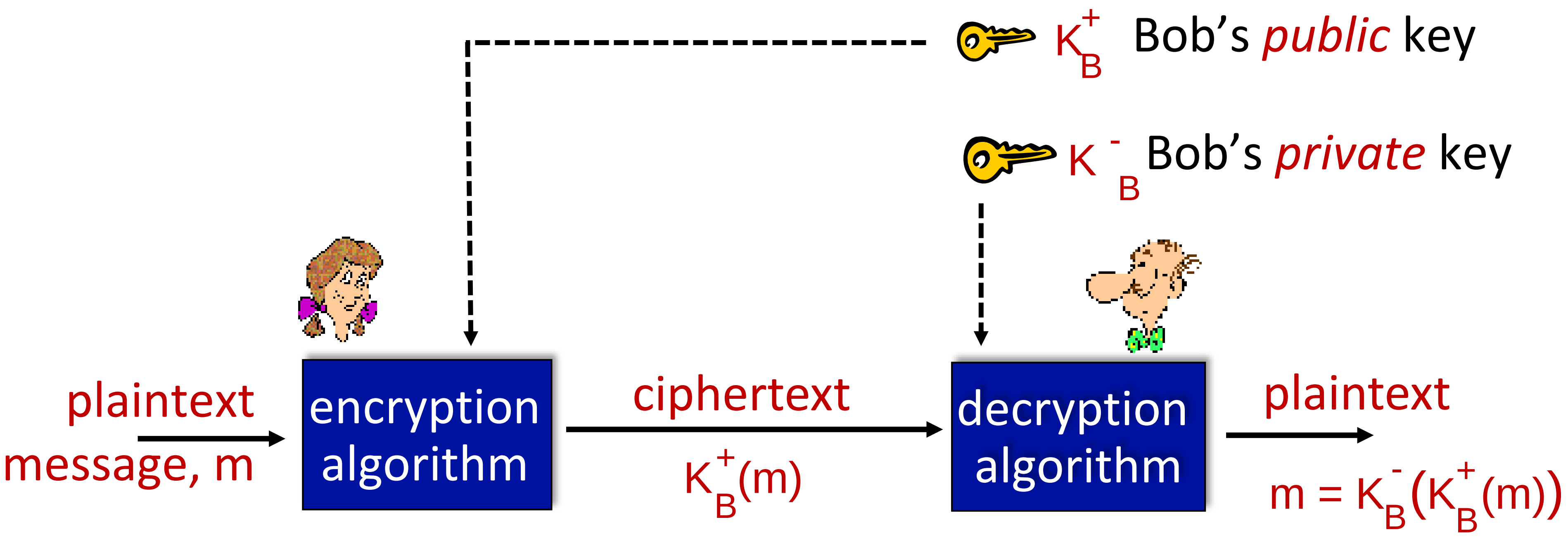


- sender, receiver do *not* share secret key
- *public* encryption key known to *all*
- *private* decryption key known only to receiver

Locking and unlocking in asymmetric-key cryptosystem



Public Key Cryptography



Public key encryption algorithms

requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

$$K_B^-(K_B^+(m)) = m$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

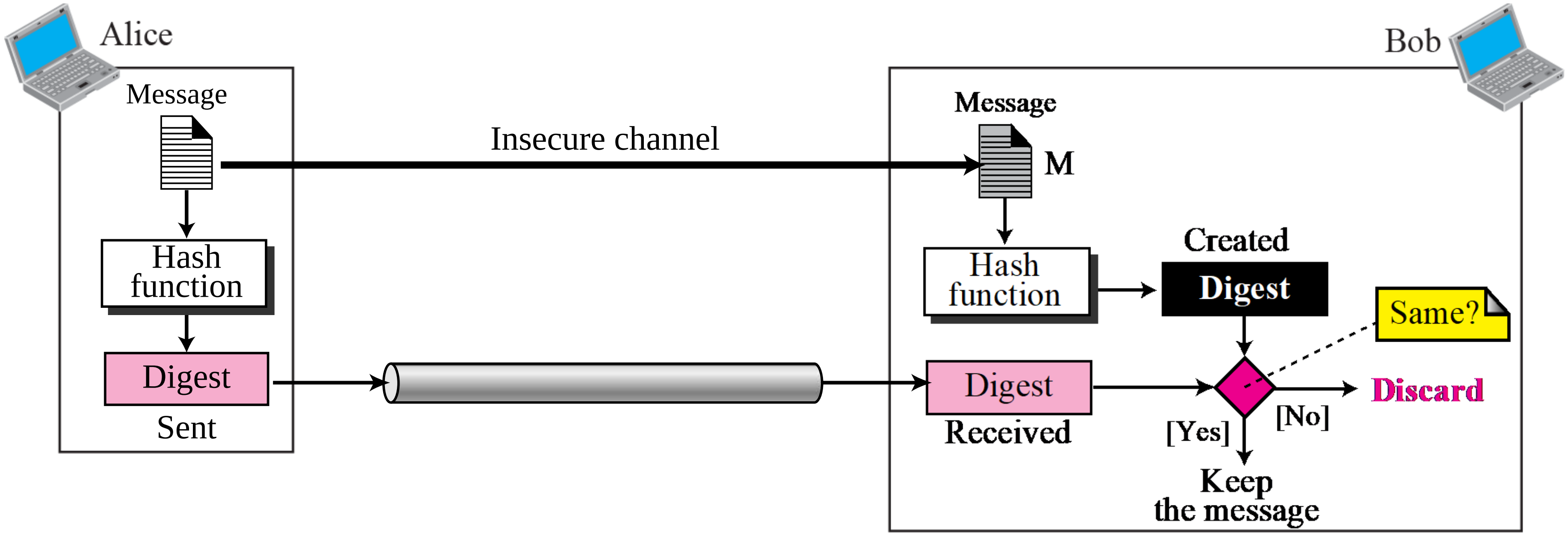
Note

In public-key cryptography, everyone has access to everyone's public key; public keys are available to the public.

Message Integrity

- The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not integrity.
- However, there are occasions where we may not even need secrecy but instead must have integrity.
- For example, Alice may write a will to distribute her estate upon her death. The will does not need to be encrypted.
- After her death, anyone can examine the will. The integrity of the will, however, needs to be preserved. Alice does not want the contents of the will to be changed.

Message and digest



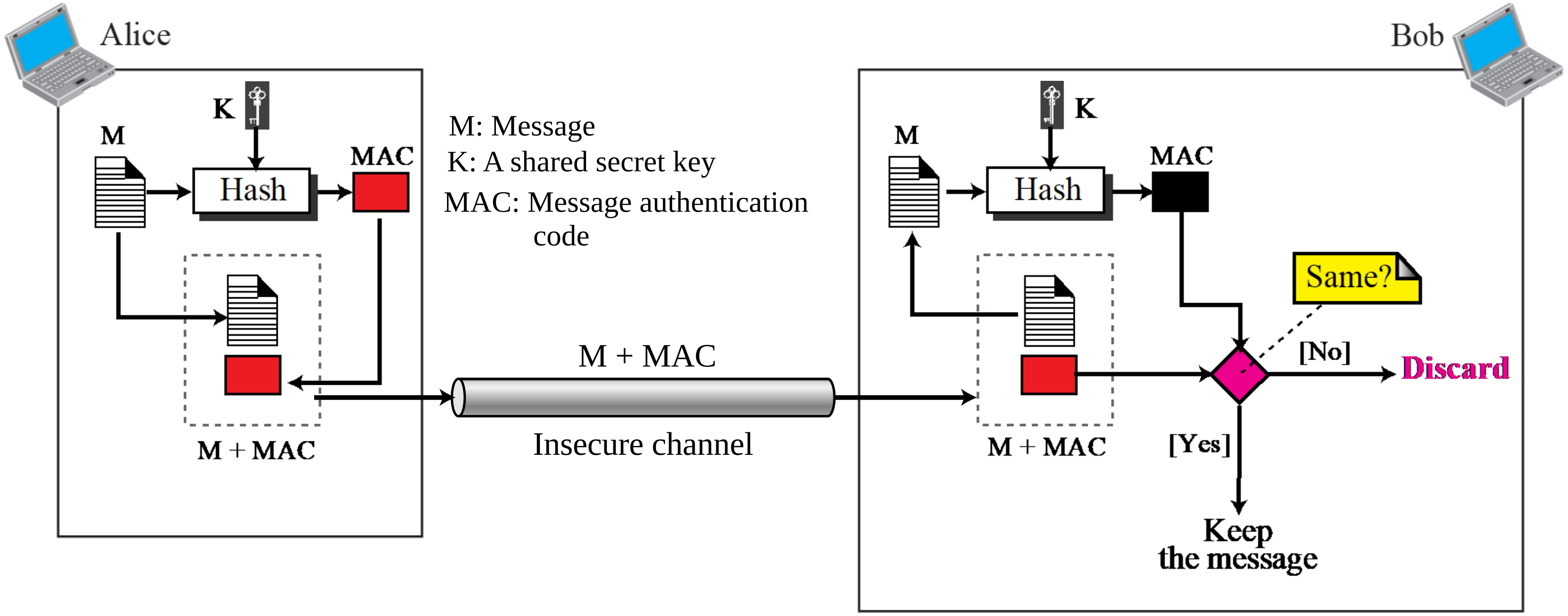
Hash function algorithms

- MD5 hash function widely used (RFC 1321)
  - computes 128-bit message digest in 4-step process.
- SHA-1 (Secure Hash Algorithm) is also used
  - US standard [NIST, FIPS PUB 180-1]
  - 160-bit message digest
- SHA-2 has four types based on output bit length
  - SHA 224 – Hash is 224 bits long.
  - SHA256
  - SHA384
  - SHA512

Message Authentication

- A digest can be used to check the integrity of a message: that the message has not been changed.

- To ensure the integrity of the message and the data origin authentication— that Alice is the originator of the message, not somebody else—we need to include a secret held by Alice in the process; we need to create a message authentication code (MAC).


Alice
Bob
K
M
MAC
Hash
M + MAC
M: Message
K: A shared secret key
MAC: Message authentication code
M + MAC
Insecure channel
K
M
MAC
Hash
Same?
[No]
Discard
M + MAC
[Yes]
Keep the message

Note

A MAC provides message integrity and message authentication using a combination of a hash function and a secret key.

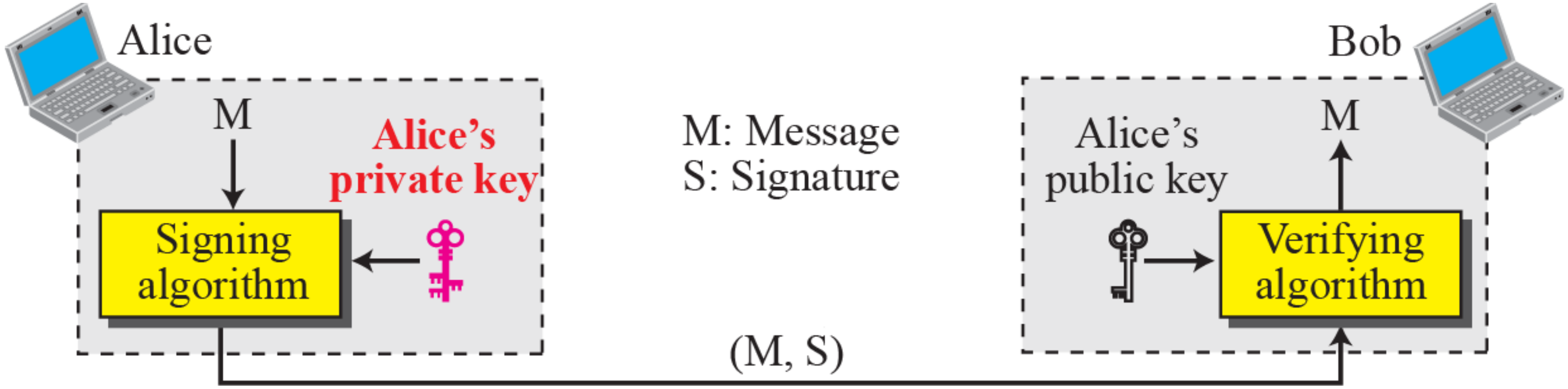
Digital Signature

Another way to provide message integrity and message authentication is a digital signature. A MAC uses a secret key to protect the digest; a digital signature uses a pair of private-public keys.

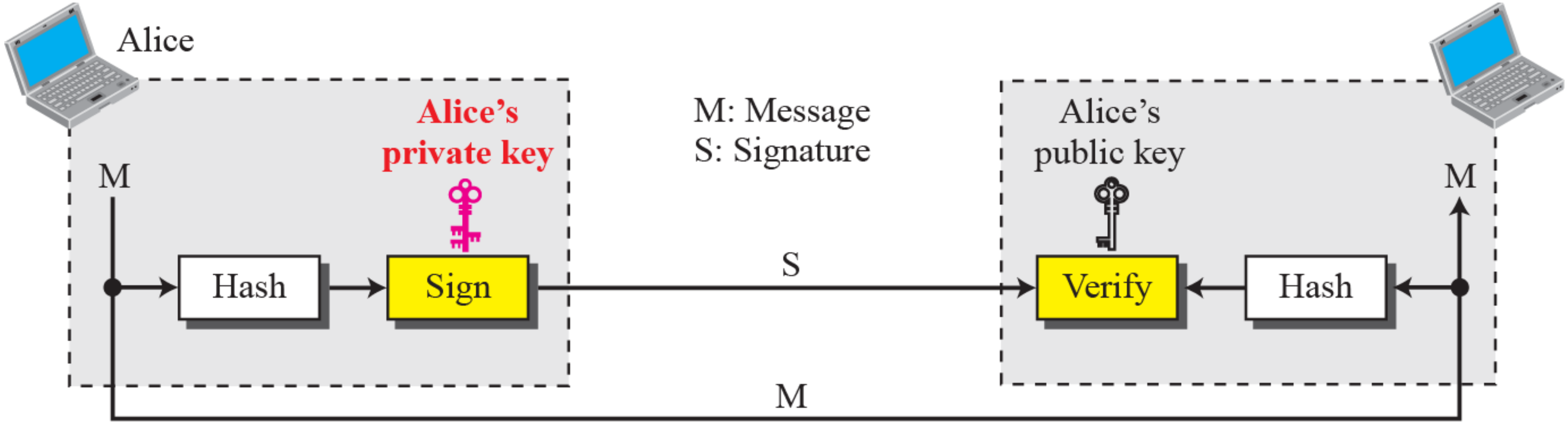
Note

A digital signature uses a pair of private-public keys.

Digital signature process

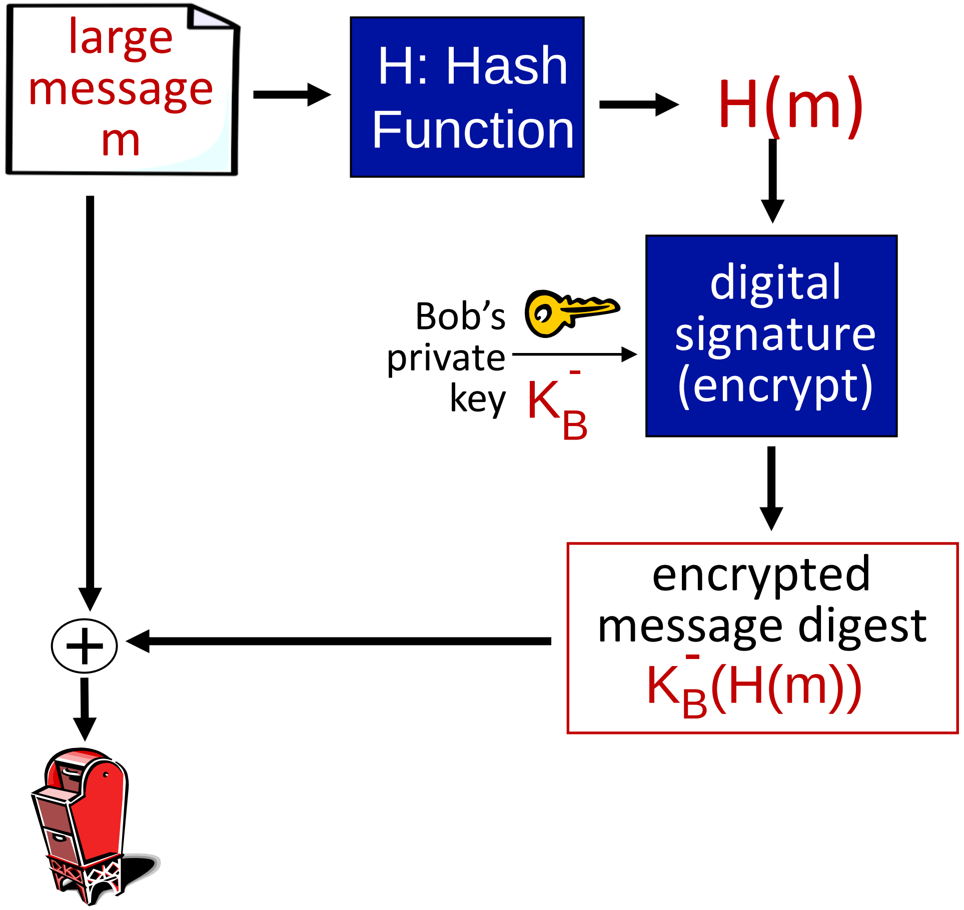


Signing the digest

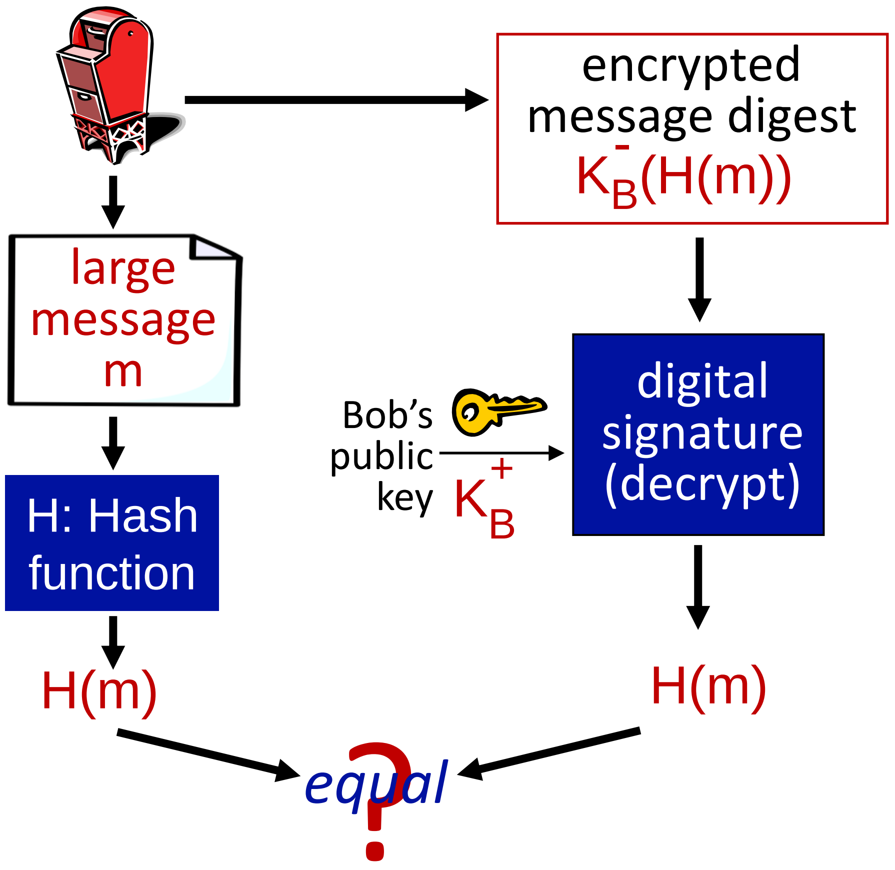


Digital signature = signed message digest

Bob sends digitally signed message:



Alice verifies signature, integrity of digitally signed message:



Note

A digital signature needs a public-key system.
The signer signs with her private key; the verifier verifies with the signer's public key.

Note

A cryptosystem uses the private and public keys of the receiver: a digital signature uses the private and public keys of the sender.

Security at different layers

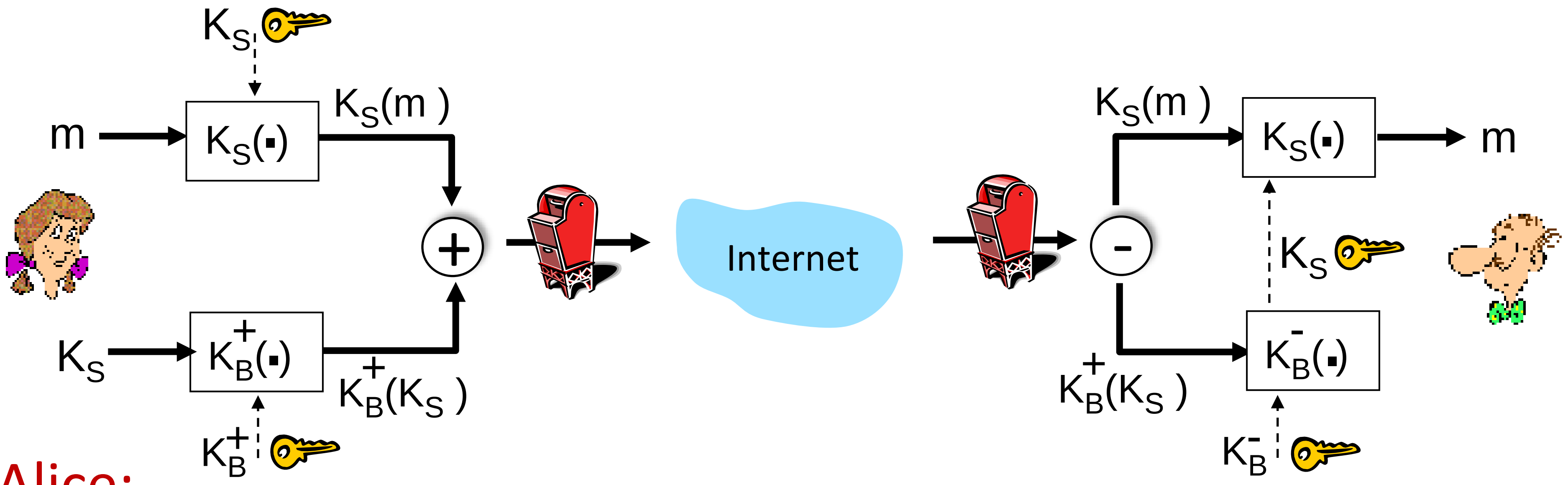
Application Layer	Pretty Good Privacy (PGP), Kerberos, Secure Shell (SSH), etc.
Transport Layer	Transport Layer Security (TLS)
Network Layer	IP Security

outline

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- Operational security: firewalls and IDS

Secure e-mail: confidentiality

Alice wants to send *confidential* e-mail, m, to Bob.

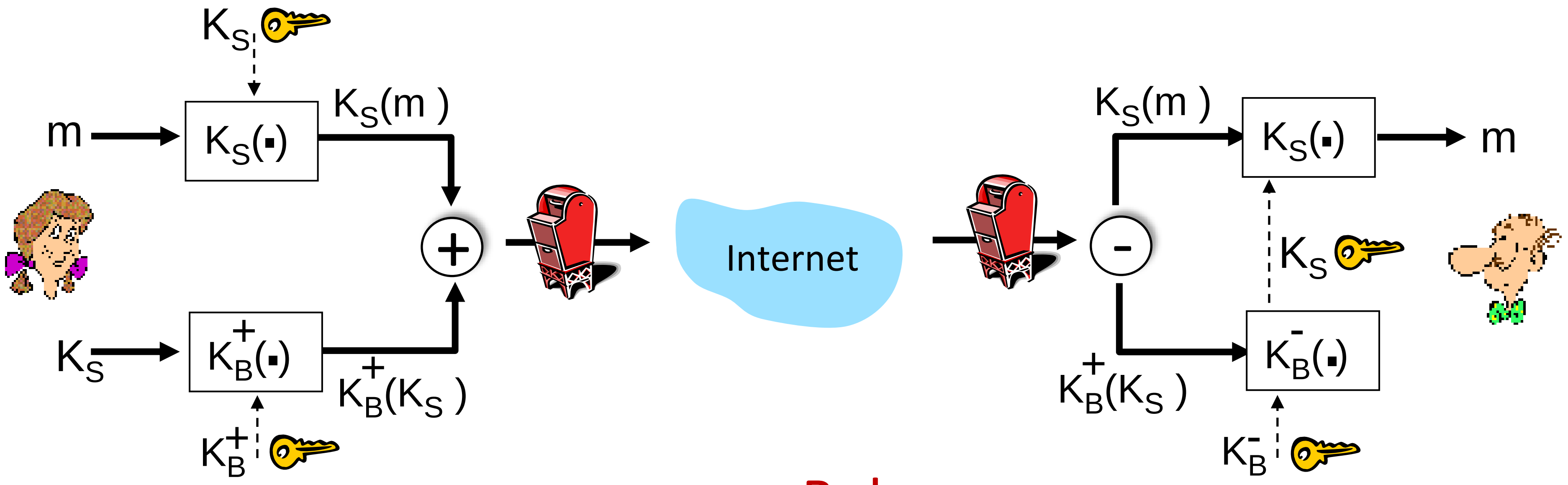


Alice:

- generates random *symmetric* private key, K_S
- encrypts message with K_S
- also encrypts K_S with Bob's public key
- sends both $K_S(m)$ and $K^+{}_B(K_S)$ to Bob

Secure e-mail: confidentiality (more)

Alice wants to send *confidential* e-mail, m, to Bob.

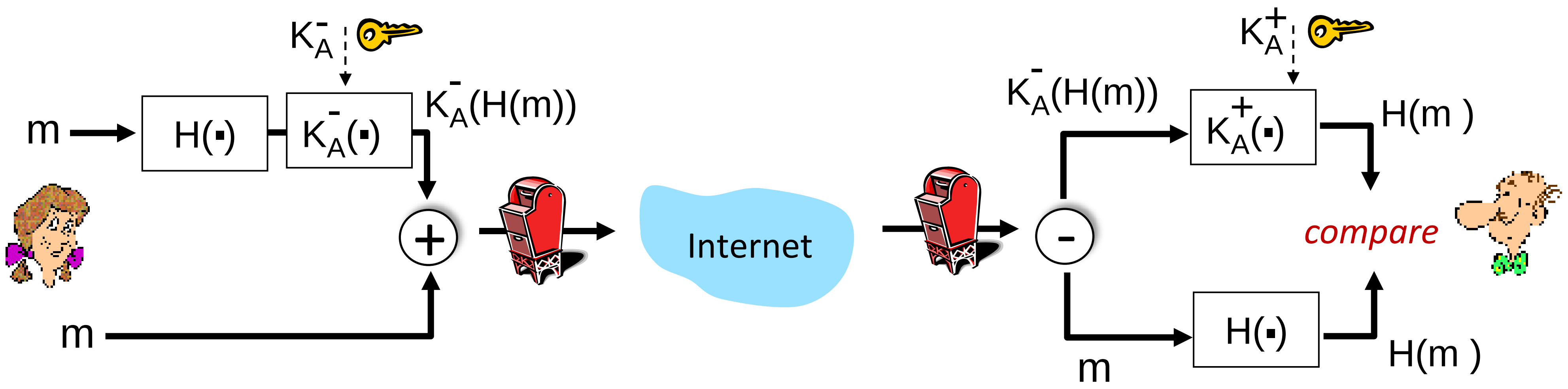


Bob:

- uses his private key to decrypt and recover K_S
- uses K_S to decrypt $K_S(m)$ to recover m

Secure e-mail: integrity, authentication

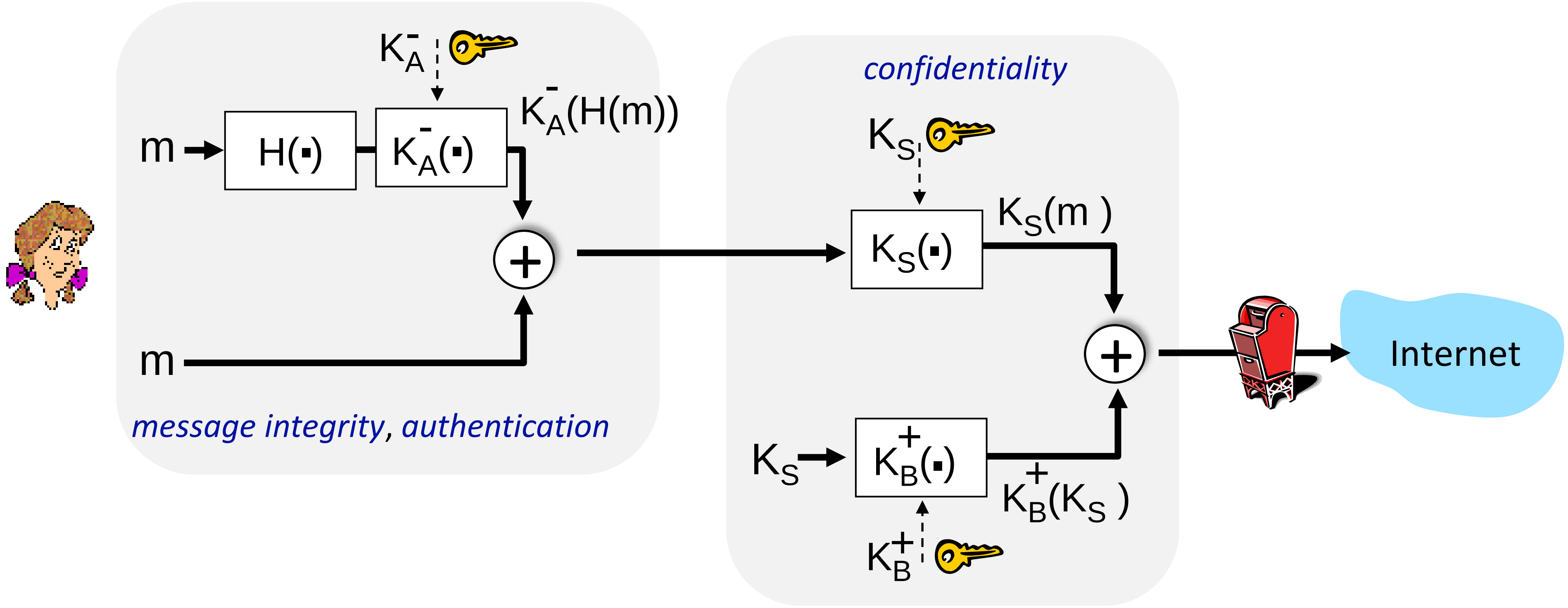
Alice wants to send m to Bob, with *message integrity*, *authentication*



- Alice digitally signs hash of her message with her private key, providing integrity and authentication
- sends both message (in the clear) and digital signature

Secure e-mail: integrity, authentication

Alice sends m to Bob, with *confidentiality, message integrity, authentication*



Alice uses three keys: her private key, Bob's public key, new symmetric key

What are Bob's complementary actions?