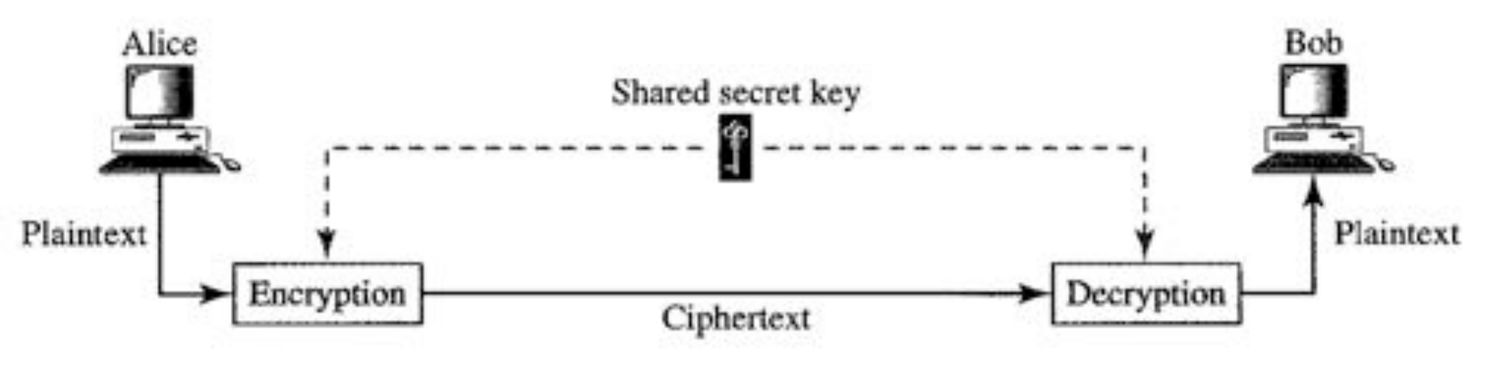
Overview

- Last Lecture
  - Data security 1 (Private key encryption)
- This Lecture
  - Data security 2 (Public key encryption)
  - Source: Sections 31.2-31.3
- Next Lecture
  - Introduction to networks
  - Source: Sections 2

Private Key Encryption

- Symmetric-key: the same key (Secret key) is used by the sender and receiver. The key is shared.
  - Caesar cipher, Transposition ciphers, DES introduced in last lecture
  - Key is very important in these methods. The secrecy of the key must be protected
  - These methods are called private key encryption



Private Key Encryption (cont.)

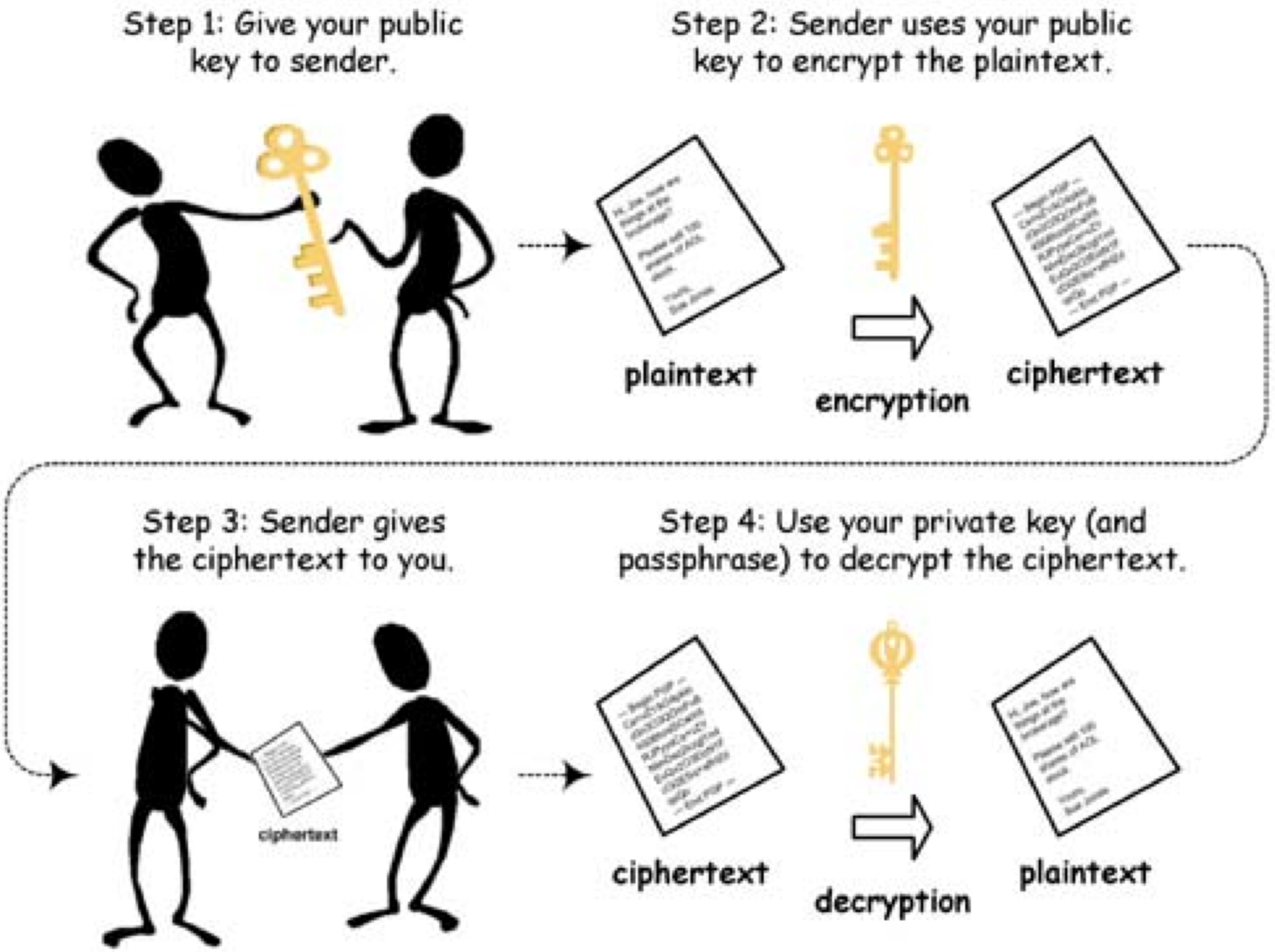
- The best encryption method in the world is no good if the key cannot be kept secret.
- How does the sender communicate the key to the receiver?
  - Sender send the key: what if an unauthorized receiver gets it?
  - Encrypt the key: same problem?
  - Can we make the encryption key public while still making the encrypted message secure?

Public Key Encryption

video1: to 1:23

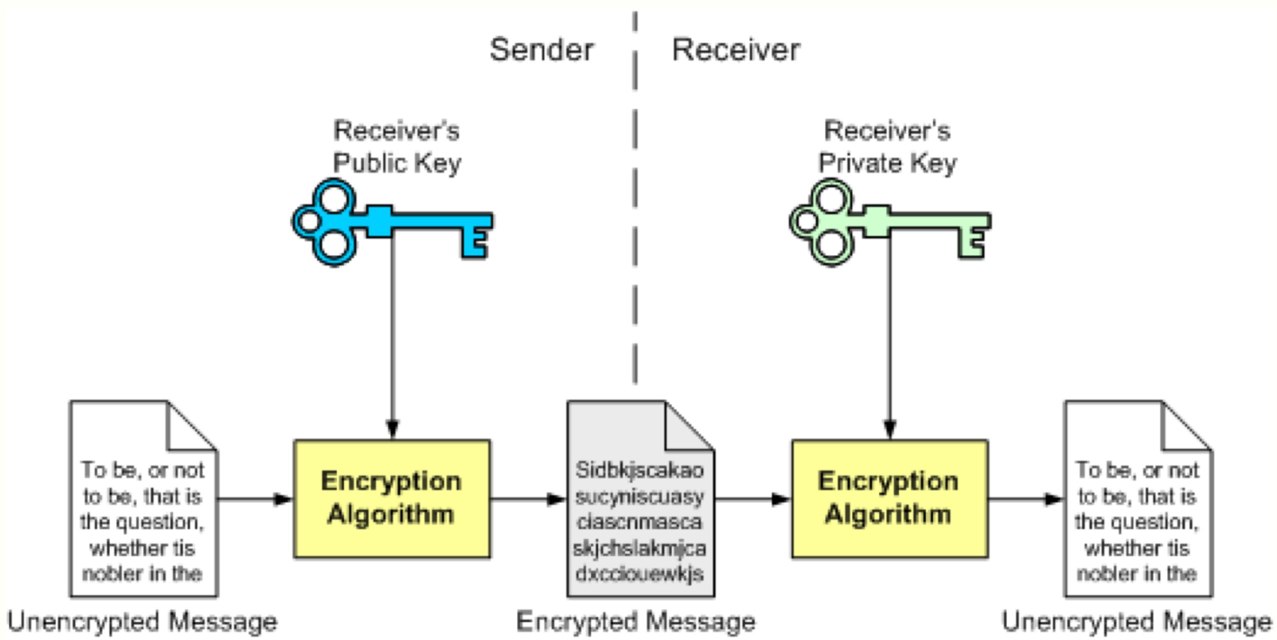
- Encryption and decryption functions are separated
- Encryption key cannot be used to decrypt a message
- The encryption key is public, but the decryption key is only known to the receiver.
- Used for banking/military use/electronic commerce

Public Key Encryption (cont.)



Public Key Encryption (cont.)

- Asymmetric-key: two separate keys instead of one
  - One key used for the sender to encrypt: <u>Public key</u>
  - One key used for the receiver to decrypt: <u>Private key</u>



http://gdp.globus.org/gt3-tutorial/multiplehtml/ch10s03.html

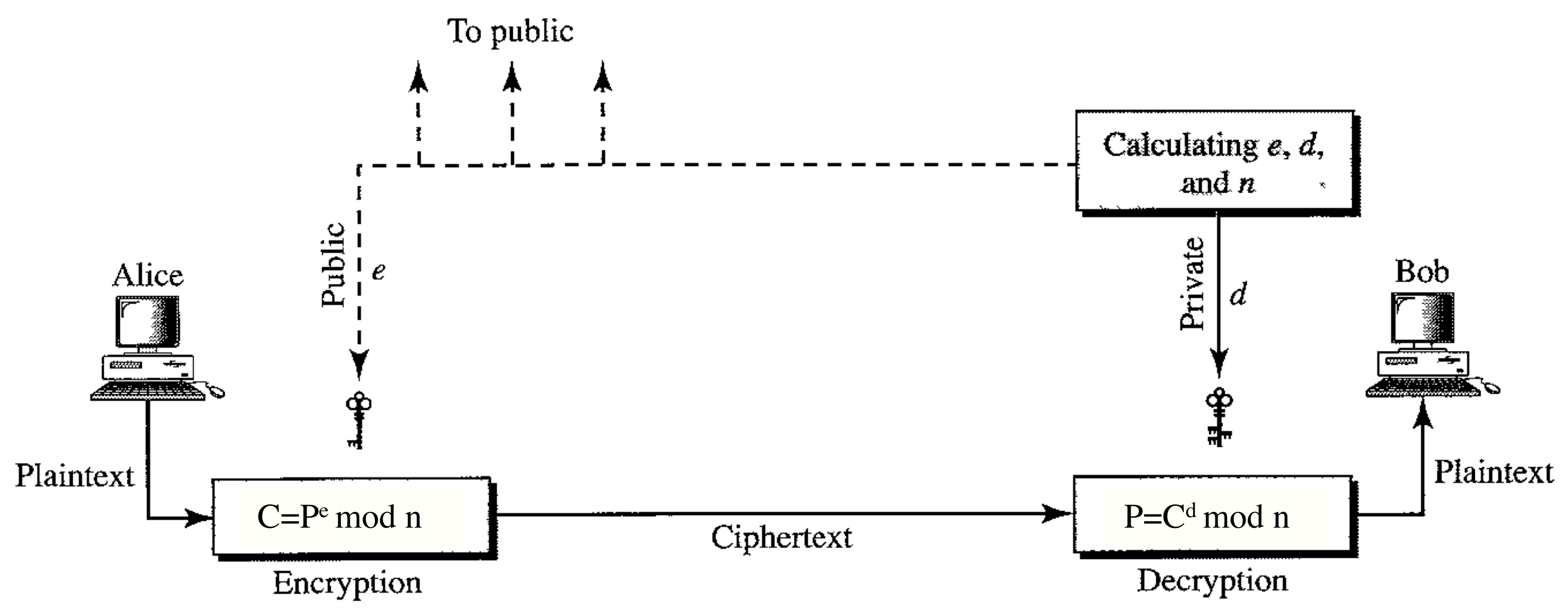
Public Key Cryptosystems (cont.)

- The most popular is called RSA algorithm, (named after its inventors: Rivest, Shamir, Adelman. Invented in 1978, Turing Award in 2002, vedio2)

Public Key Cryptosystems (cont.)

- Terms used in RSA algorithm
  - prime number: number that can only be divided by 1 or itself
  - Greatest Common Divisor (GCD): largest number that two numbers can be divided by
  - Relatively Prime: when the GCD of two numbers is 1
  - Modulo: operation that gives the remainder of a division

Public Key Cryptography algorithms



The two keys, public key e and private key d, have a special relationship to each other! (number theory)

RSA Algorithm

video 3

- Bob use the following steps to select private and public keys:
  - Choose two prime numbers, p and q. p = 7; q = 11
  - $n = p \times q$ $n = 7 \times 11 = 77$
  - $m = (p-1) \times (q-1)$ $m = 6 \times 10 = 60$
  - Select e that is relatively prime to m. (That is, the greatest common divisor of e and m is 1.) e = 7
  - Find d such that $(e \times d - 1) \bmod m = 0$ d= 43
    - This means $e \times d - 1$ is evenly divisible by m, or $d \times e = k \times m + 1$, $d = (k \times m + 1)/e$
- Public key = {e} (Bob announces e=7 and n=77 to the public)
- Private key = {d} (Bob keeps d=43 and m=60 secret)
- Encryption: Ciphertext, $C = P^e \bmod n$
  - (Alice calculates using public key e=7, $C = 8^7 \bmod 77 = 57$)
- Decryption: Plaintext, $P = C^d \bmod n$
  - (Bob calculates using private key d=43, $P = 57^{43} \bmod 77 = 8$)

Example

- Public key: 7, Private key: 43
- Suppose HELLO is to be sent (8, 5, 12, 12, 15)
- 8^7 mod 77 = 57; 5^7 mod 77 = 47; 12^7 mod 77 = 12; 15^7 mod 77 = 71
- Send 57, 47, 12, 12, 71
- 57^{43} mod 77 = 8; 47^{43} mod 77 = 5; 12^{43} mod 77 = 12; 71^{43} mod 77 = 15

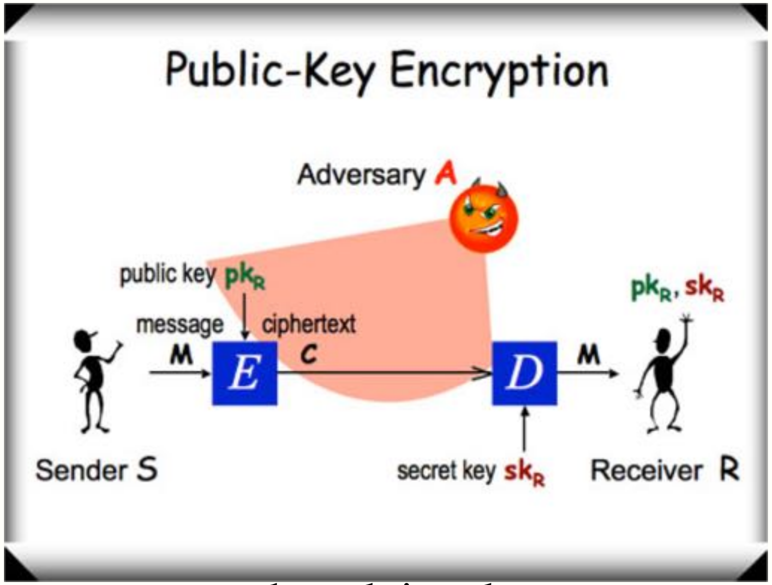
The encryption and decryption are surprisingly simple! – exponentiation and modular arithmetic

Online demo: http://cobweb.cs.uga.edu/~dme/csci6300/Encryption/Crypto.html

RSA Algorithm (cont.)

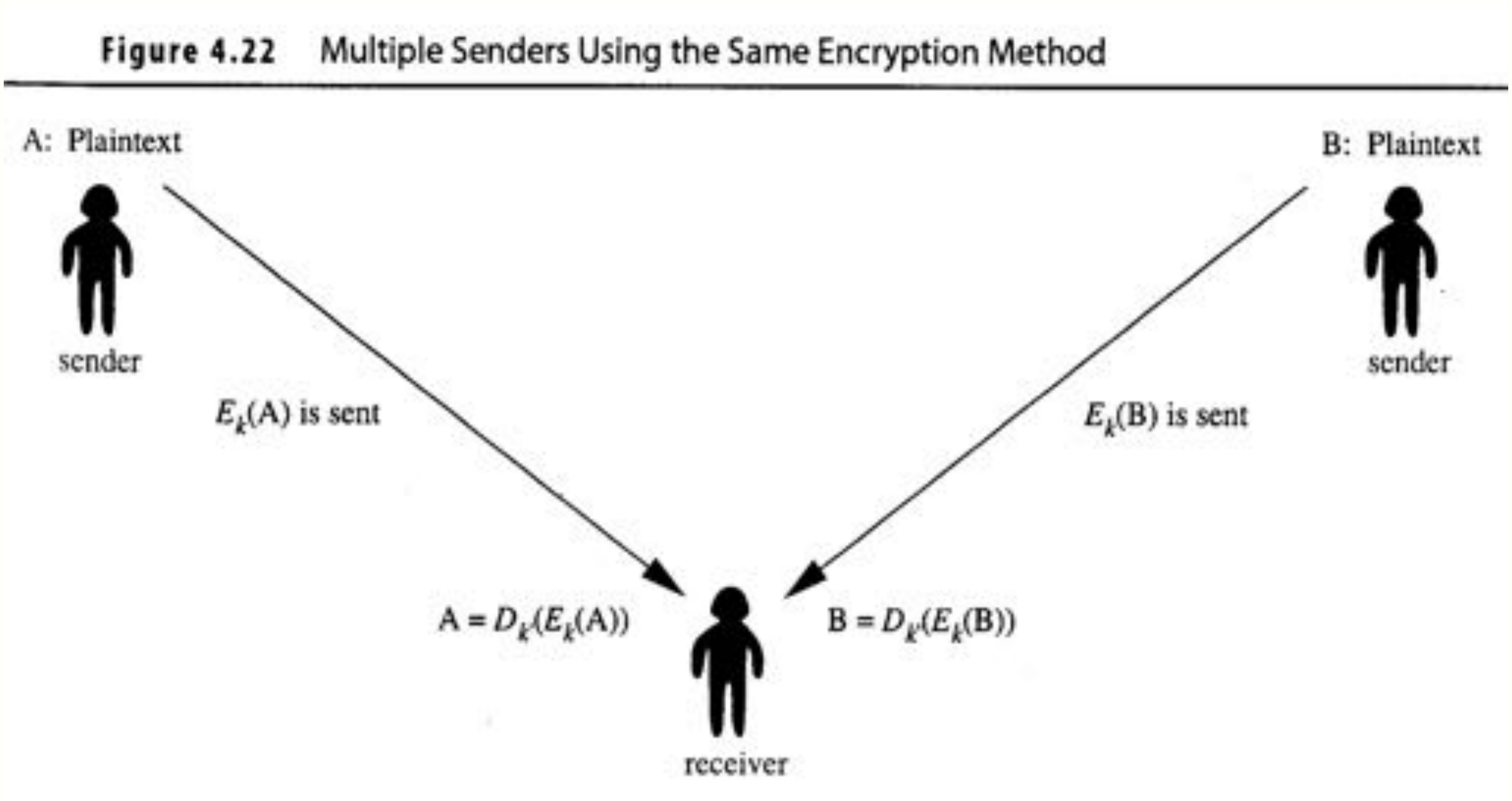
- Justification of security
  - Suppose you know *e* and *n*, and intercept the encrypted message.
  - Since the encryption is not reversible, you cannot decrypt the message by a reverse operation.
  - You would need to find *p* and *q*, where $p \times q = n$, and then find *d*, where $e \times d \mod (p-1)*(q-1) = 1$
  - If *n* is very large (e.g. 159,197), finding *p* and *q* that are prime numbers (factors of *n*) is very time consuming.

The ciphertext is very difficult to break even when public key e and n are known.



bowdoin.edu

Public Key Encryption (cont.)
video 4



Public Key Encryption (cont.)

- Problem: RSA is slow if the message is long
- RSA is useful for short messages, such as small message digest, a symmetric key etc.
- RSA Applications:
  - Authentication (Digital signature)
    - Signing the Message
    - Signing the Message Digest
  - PGP

Authentication

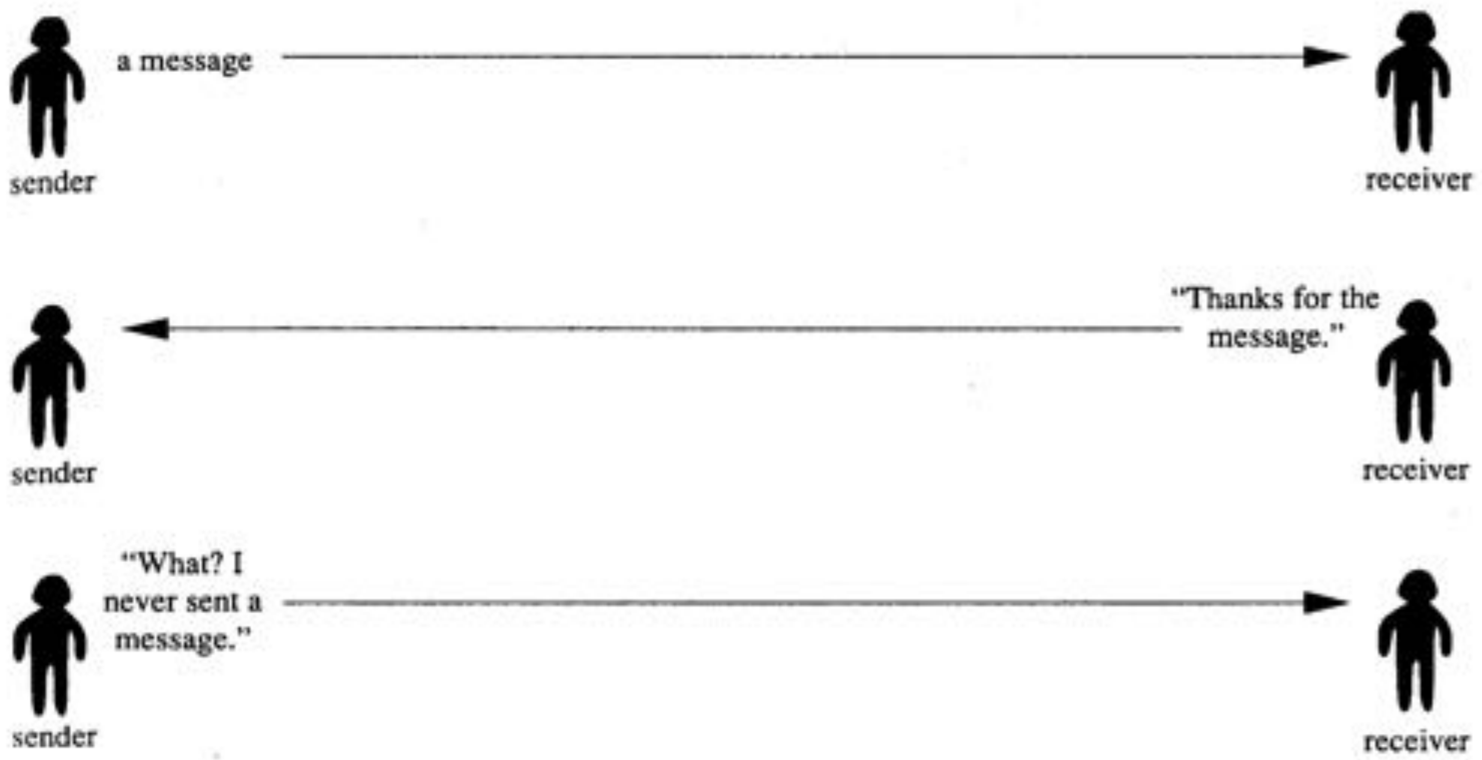


Figure 4.23 Sender Denying Sending a Message

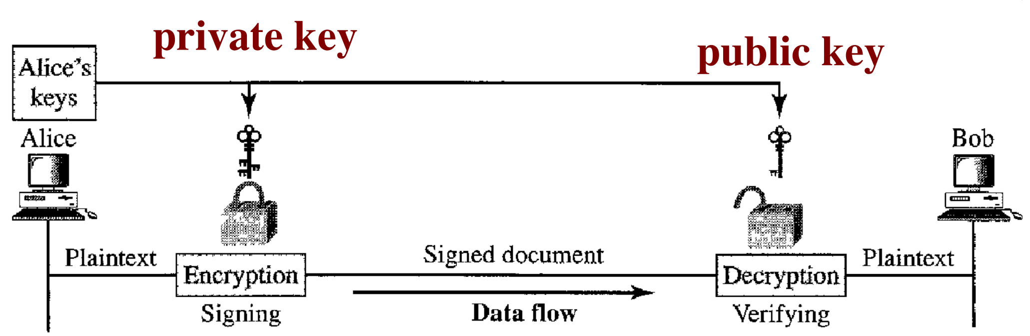
Authentication (cont.)

- Verifying the identity of a sender is called authentication
  - The sender may deny his/her message
  - Somebody may pretend to be the sender
- It is very important for banking system, e-commerce, digital contracts.

Authentication (cont.)

- One method of authentication: digital signature
- Signing the Message (Document) using the private and public key of the sender, instead of the receiver

Figure 31.11 *Signing the message itself in digital signature*



  - Since *only* the sender's public key can decrypt the digital signature (encrypted with the sender's *private* key), Alice is the only person who sent the message.

Authentication (cont.)

- Signing the Message Digest
  - Public key is inefficient to dealt with long messages
  - Want to know that an electronic document or file has not be altered
    - Document: Message
    - Fingerprint: Message digest
    - hash function

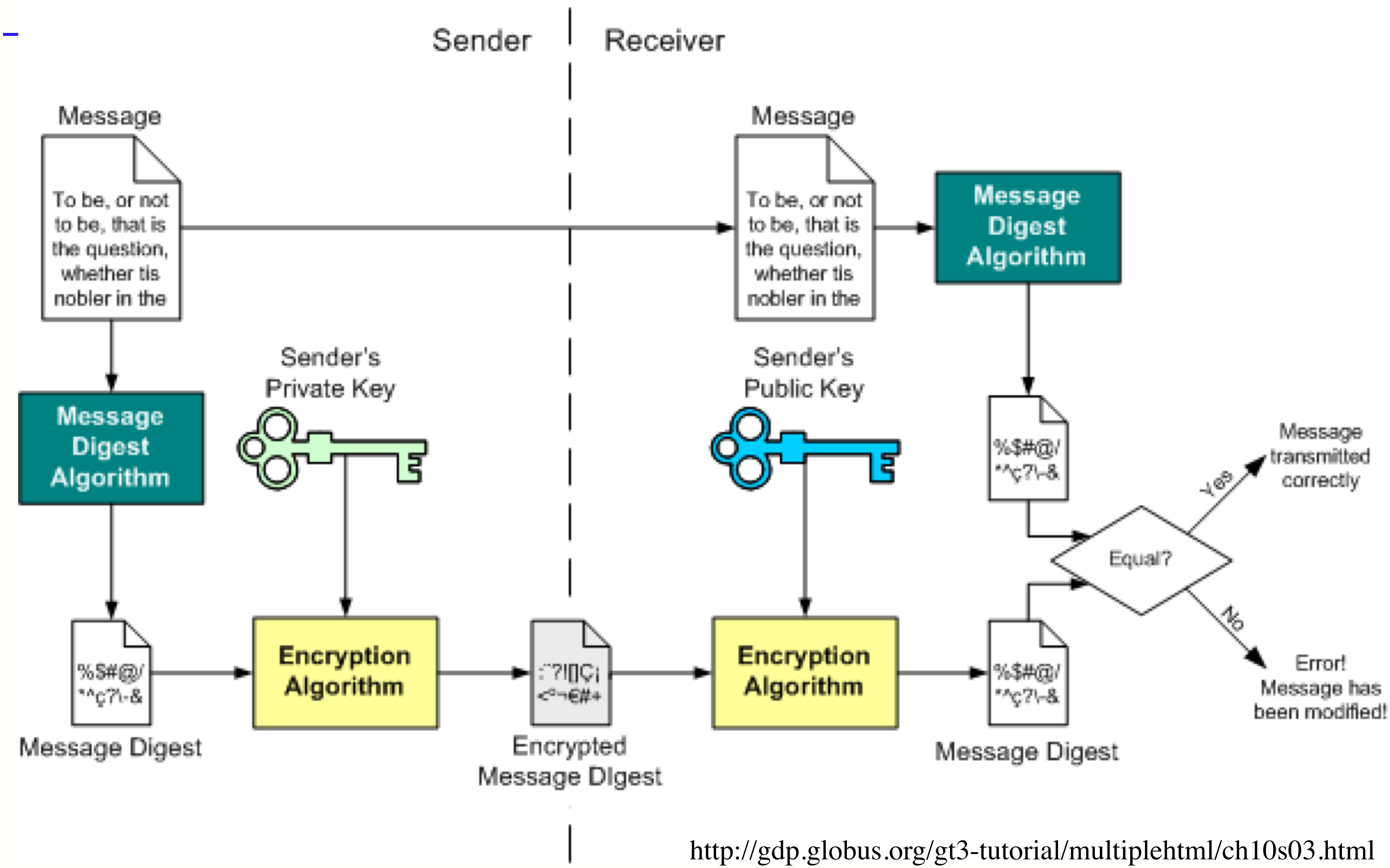
Authentication (cont.)

- Sender does the following:
  - Step 1: A *message digest* is generated. ('summary' of the message)
    - smaller than the message itself
    - slightest change in the message produces a different digest.
    - generated using a set of hashing algorithms
  - Step 2: The message digest is encrypted using the sender's *private* key. (The resulting encrypted message digest is the digital signature.)
  - Step 3: The digital signature is attached to the message, and sent to the receiver.

Authentication (cont.)

- The receiver then does the following:
  - Using the sender's public key, decrypts the digital signature to obtain the message digest generated by the sender.
  - Uses the same message digest algorithm used by the sender to generate a message digest of the received message.
  - Compares both message digests (the one sent by the sender as a digital signature, and the one generated by the receiver). If they are not *exactly the same*, the message has been tampered with by a third party

Authentication (cont.)



http://gdp.globus.org/gt3-tutorial/multiplehtml/ch10s03.html

PGP

- Pretty Good Privacy (PGP): encryption program developed by Phil Zimmerman in 1991. (free, email security)
- a combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography
  - The message is encrypted using a symmetric-key encryption algorithm, which requires a symmetric key. (Each symmetric key is used only once and is also called a session key.)
  - The session key is encrypted using public-key encryption
  - The encrypted message along with the encrypted session key is sent to the receiver.
- using the strengths of one algorithm to compensate for the weaknesses of the other, PGP is one of the strongest and fastest encrypting algorithm.

Summary

- Symmetric-key and public-key
- Public key systems
  - RSA algorithm
  - Authentication methods
  - Digital signature
  - Hash-based authentication
- PGP