# ASYMMETRIC CRYPTOGRAPHIC ALGORITHMS

# Problem With Symmetric Key

- ***Problem !***
- Suppose sender & receiver may be in different countries.
- ✓ E.g:- Online shopping website
- ❖ How they will exchange the key & agree on it?
- ✓ Physically visit
- ✓ Courier

Not Convenient

- ✓ Internet & ask for confirmation.
- **If Intruder gets the key, he can unlock the things.**
- ***Problem 2***
- Separate/Unique key for each communication is needed.
- ✓ E.g:- A to B & A to C or B to C
- To overcome Interruption Attack

# Public-Key Cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody can be freely distributed, 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

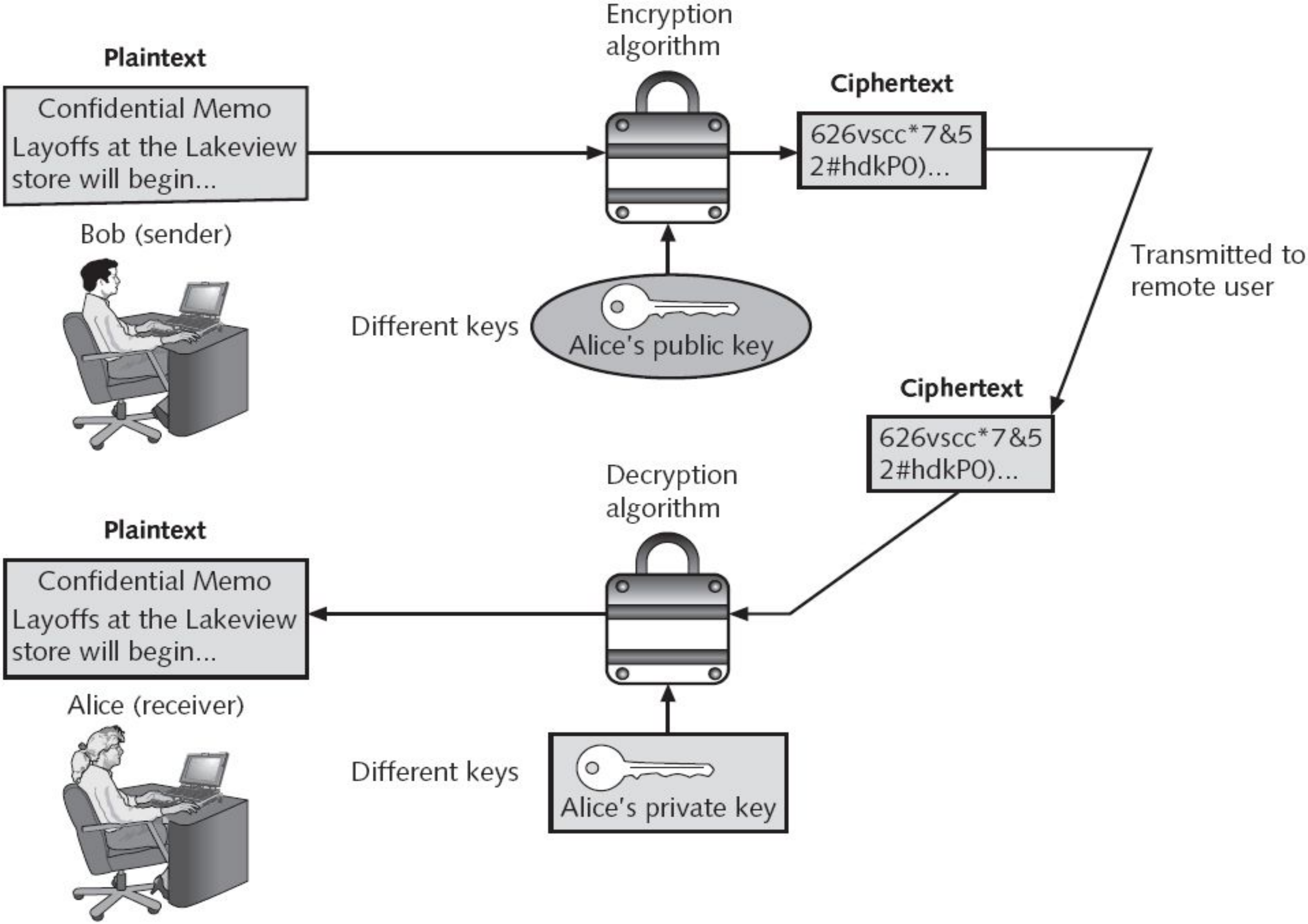



**Figure 11-12** Asymmetric cryptography

# DIFFIE-HELLMAN KEY EXCHANGE/AGREEMENT ALGORITHM

- Introduction
- Description of the algorithm
- Example of the algorithm
- Mathematical theory behind the algorithm
- Problems with the algorithm

# Father of AKC

- In the mid- 1970’s , Whitefield Diffie ,a student at the Stanford University met with Martin Hellman, his professor &the two began to think about it.
- After some research & complicated mathematical analysis, they came up with the idea of AKC.
- Many experts believe that this development is the first & perhaps the only truly revolutionary concept in the history of cryptography

# Diffie-Hellman

- Developed to address shortfalls of *key distribution* in symmetric key distribution.
- A *key exchange algorithm*, not an encryption algorithm
- Allows two users to share a *secret key* securely over a public network
- Once the key has been shared
  - Then both parties can use it to encrypt and decrypt messages using symmetric cryptography

# Diffie Hellman

- Algorithm is based on “difficulty of calculating discrete logarithms in a finite field”
- ***These keys are <u>mathematically</u> related to each other.***
- ***‘’Using the public key of users, the session key is generated without transmitting the private key of the users.’’***
- Vulnerable to “man in the middle” attacks*

# DIFFIE-HELLMAN KEY EXCHANGE/AGREEMENT ALGORITHM

1. Firstly, Alice and Bob agree on two large prime numbers, n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.

   Let n = 11, g = 7.

2. Alice chooses another large random number x, and calculates A such that:
   $A = g^x \bmod n$

   Let x = 3. Then, we have, $A = 7^3 \bmod 11 = 343 \bmod 11 = 2$.

3. Alice sends the number A to Bob.

   Alice sends 2 to Bob.

4. Bob independently chooses another large random integer y and calculates B such that:
   $B = g^y \bmod n$

   Let y = 6. Then, we have, $B = 7^6 \bmod 11 = 117649 \bmod 11 = 4$.

5. Bob sends the number B to Alice.

   Bob sends 4 to Alice.

6. A now computes the secret key K1 as follows:
   $K1 = B^x \bmod n$

   We have, $K1 = 4^3 \bmod 11 = 64 \bmod 11 = 9$.

7. B now computes the secret key K2 as follows:
   $K2 = A^y \bmod n$

   We have, $K2 = 2^6 \bmod 11 = 64 \bmod 11 = 9$.

# Diffie-Hellman Key exchange

- Public values:
  - large prime p, generator g (primitive root of p)
- Alice has secret value x, Bob has secret y
- Discrete logarithm problem: given x, g, and n, find A
- $A \rightarrow B: g^x \pmod n$
- $B \rightarrow A: g^y \pmod n$
- Bob computes $(g^x)^y = g^{xy} \pmod n$
- Alice computes $(g^y)^x = g^{xy} \pmod n$

# man-in-the-middle attack

| Alice | Tom | Bob |
|---|---|---|
| $n = 11, g = 7$ | $n = 11, g = 7$ | $n = 11, g = 7$ |

man-in-the-middle attack **Part-I**

# man-in-the-middle attack

| | | |
|---|---|---|
| Alice | Tom | Bob |
| $x = 3$ | $x = 8$, $y = 6$ | $y = 9$ |

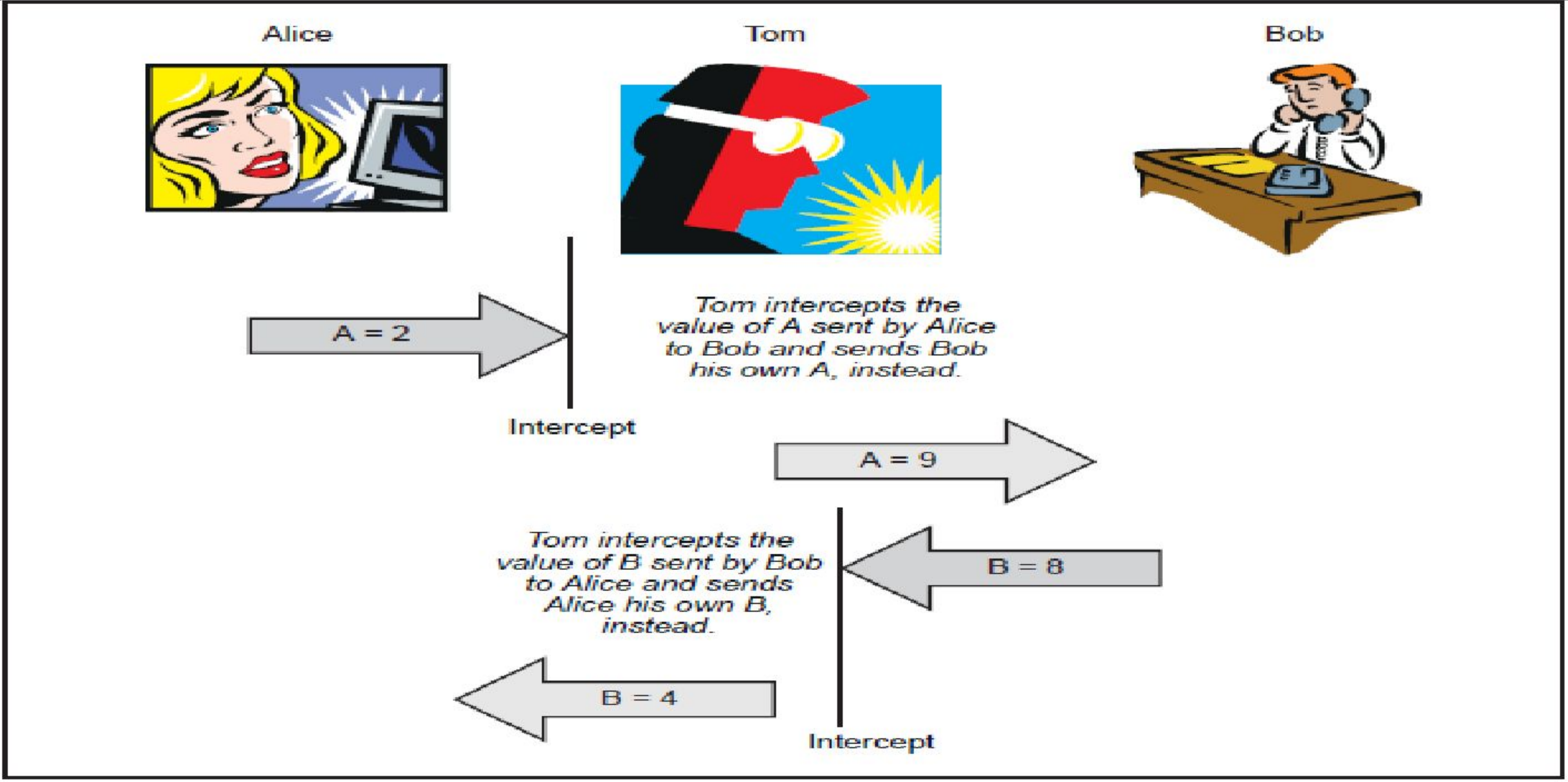
man-in-the-middle attack **Part-II**

# man-in-the-middle attack

| Alice | Tom | Bob |
|---|---|---|
| $A = g^x \bmod n$ | $A = g^x \bmod n$ | $B = g^y \bmod n$ |
| $= 7^3 \bmod 11$ | $= 7^8 \bmod 11$ | $= 7^9 \bmod 11$ |
| $= 343 \bmod 11$ | $= 5764801 \bmod 11$ | $= 40353607 \bmod 11$ |
| $= 2$ | $= 9$ | $= 8$ |
| | $B = g^y \bmod n$ | |
| | $= 7^6 \bmod 11$ | |
| | $= 117649 \bmod 11$ | |
| | $= 4$ | |

man-in-the-middle attack **Part-III**

# man-in-the-middle attack




man-in-the-middle attack **Part-IV**

# man-in-the-middle attack

| Alice | Tom | Bob |
|---|---|---|
| A = 2, B = 4* | A = 2, B = 8 | A = 9*, B = 8 |

(Note: * indicates that these are the values after Tom hijacked and changed them.)

man-in-the-middle attack **Part-V**

# man-in-the-middle attack

Alice

$K1 = B^x \bmod n$

$= 4^3 \bmod 11$

$= 64 \bmod 11$

$= 9$

Tom

$K1 = B^x \bmod n$

$= 8^8 \bmod 11$

$= 16777216 \bmod 11$

$= 5$

$K2 = A^y \bmod n$

$= 2^6 \bmod 11$

$= 64 \bmod 11$

$= 9$

Bob

$K2 = A^y \bmod n$

$= 9^9 \bmod 11$

$= 387420489 \bmod 11$

$= 5$

man-in-the-middle attack **Part-VI**

# Preventing a Man-in-the-Middle Attack with Hashing

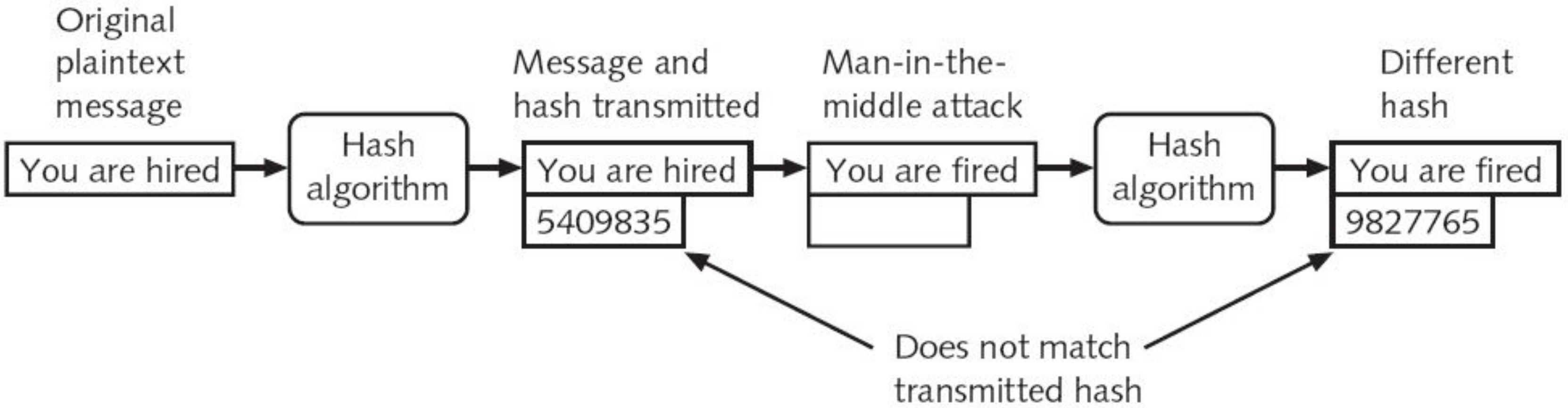



**Figure 11-4** Man-in-the-middle attack defeated by hashing