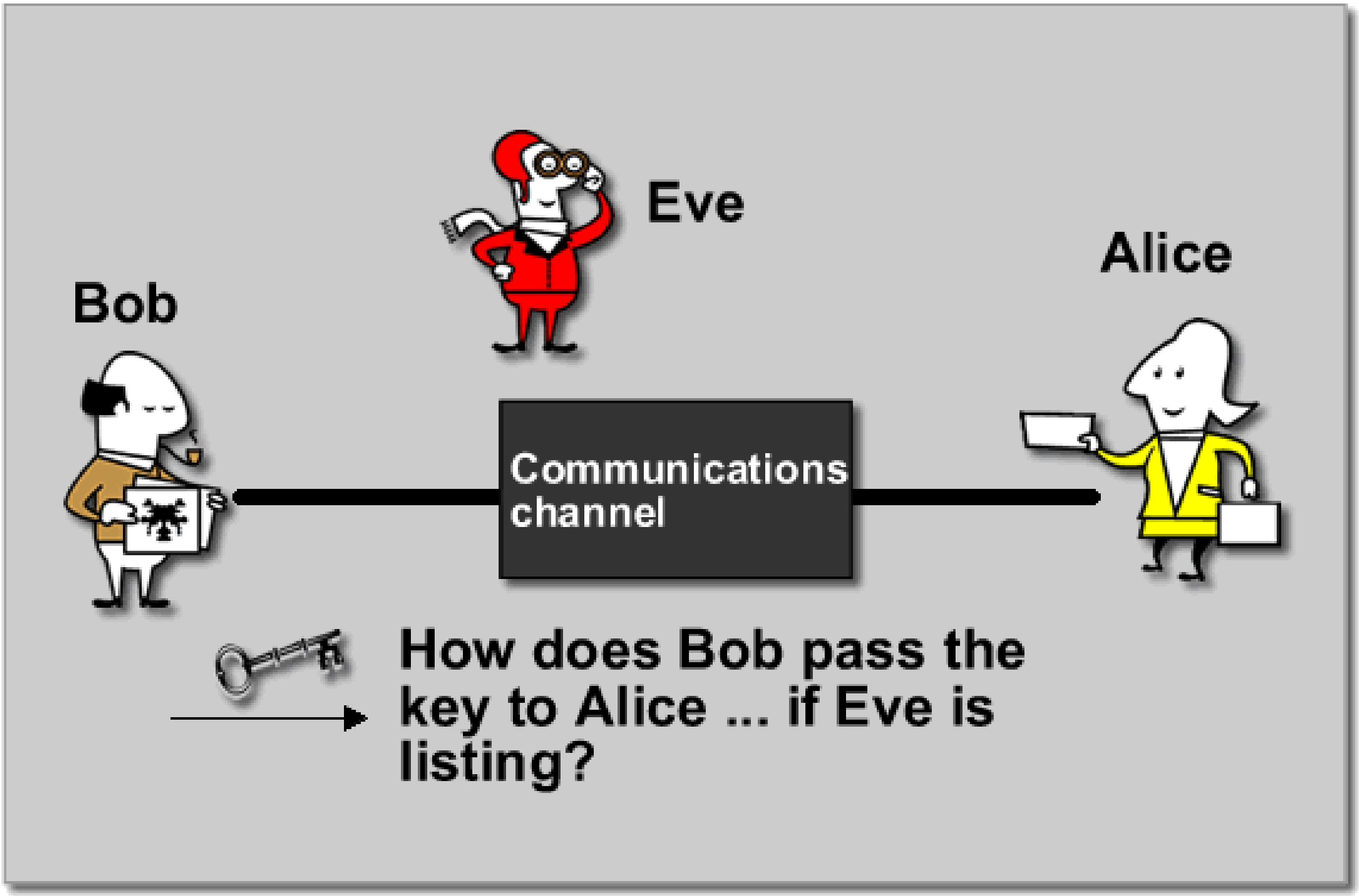
Diffie-Hellman Key Exchange

Key Establishment: The problem

- Securing communication requires that the data is encrypted before being transmitted.
- Associated with encryption and decryption are keys that must be shared by the participants.
- The problem of securing the data then becomes the problem of securing the establishment of keys.
- Task: If the participants do not physically meet, then how do the participants establish a shared key?

Key Establishment: The problem (cont.)



Diffie-Hellman Key Exchange

- Discovered by Whitfield Diffie and Martin Hellman
  - "New Directions in Cryptography" (1976)

- Diffie-Hellman key Exchange protocol
  - Exponential key agreement
  - Allows two users to exchange a secret key
  - Requires no prior secrets
  - Real-time over an untrusted network

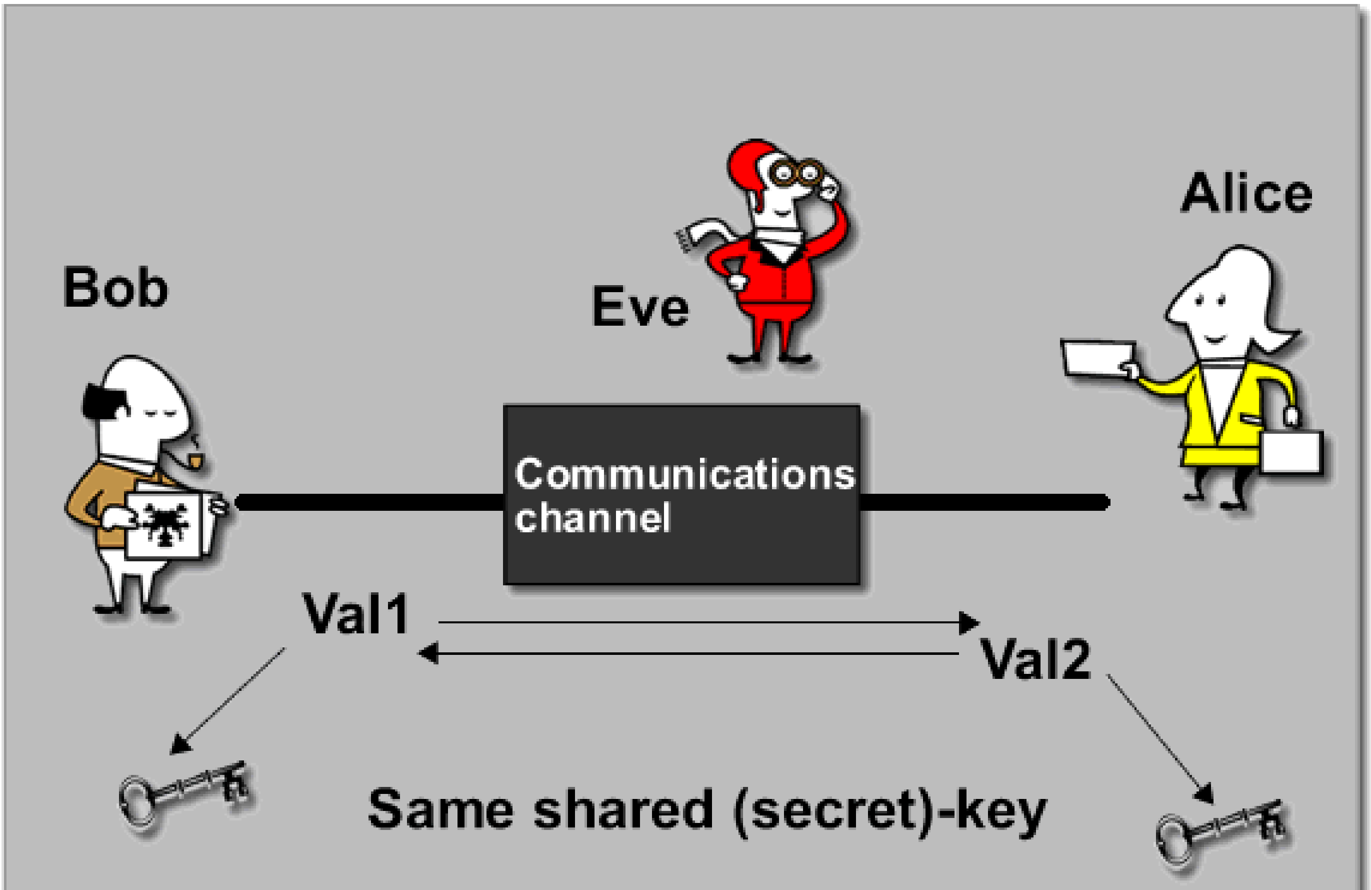
Diffie-Hellman Key Exchange (cont..)

- Using Diffie-Hellman key exchange protocol,
  - Two unknown users can set up a private but random key for their symmetric key cryptosystem.

- There is no need for users to meet in advance, or use a secure courier, or use some other secret channels, to select a key.

- The purpose of the algorithm is exchange of a secret key
  - Not encryption
  - Not signing

Diffie-Hellman Key Exchange (cont..)



Algorithm

- Require two large numbers,
  - one prime p,
  - and generator g (2 <= g <= p-2), is a primitive root of p,
- p and g both are publicly available numbers

Primitive Root

- a primitive root of a prime number p as one whose powers modulo p generate all the integers from 1 to p-1. That is, if a is a primitive root of the prime number p, then the numbers
  - $a \bmod p, a^2 \bmod p, \ldots, a^{p-1} \bmod p$
  - are distinct and consist of the integers from 1 through p-1 in some permutation (in any order)
  - g is Primitive root, it must be : 2 <= g <= p-2

Primitive Root

- P = 7
  - then 3 is primitive root of 7
  - 2 is not primitive root of 7

n (n goes to 1 to p-1)	3^n	3^n mod 7
1	3	3
2	9	2
3	27	6
4	81	4
5	243	5
6	729	1

n (n goes to 1 to p-1)	2^n	2^n mod 7
1	2	2
2	4	4
3	8	1
4	16	2
5	32	4
6	64	1

Primitive Root

- The order of $Z_n^{\times}$ is given by Euler's totient function Φ(n).

- The totient Φ(n) of a positive integer *n* is defined to be the number of positive integers less than or equal to *n* that are co-prime to *n*.
  - Example:- Φ(9) = 6, How?

Primitive Root

- Take for example n = 14. The elements of $\mathbf{Z}_{14}^{\times}$ are the congruence classes {1, 3, 5, 9, 11, 13}; there are φ(14) = 6 of them.
- Here is a table of their powers (mod 14):

n   n, n^2, n^3, ... (mod 14)

1 : 1,

3 : 3, 9, 13, 11, 5, 1

5 : 5, 11, 13, 9, 3, 1

9 : 9, 11, 1

11 : 11, 9, 1

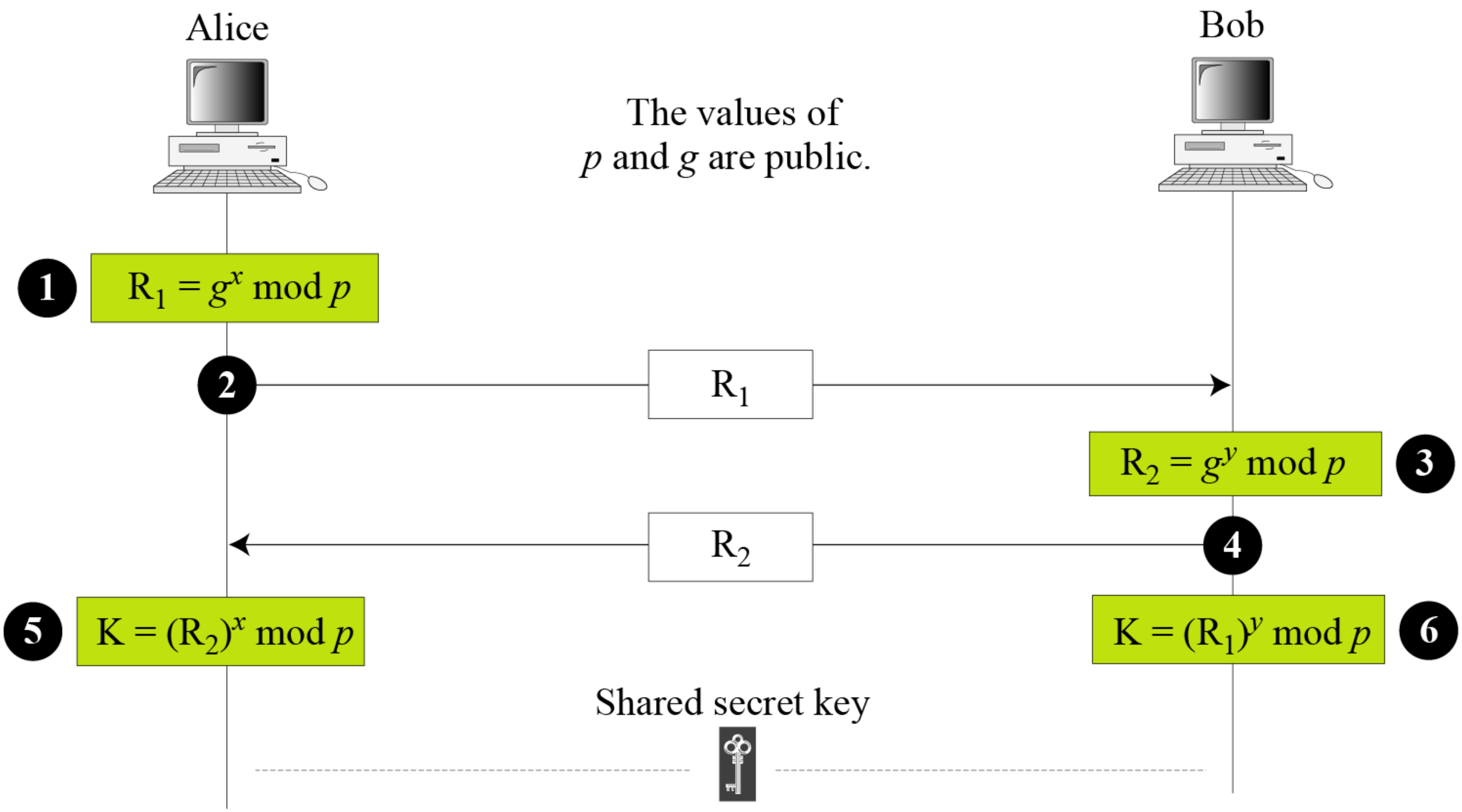
13 : 13, 1

- 3 and 5 are the primitive roots modulo 14

Algorithm

- Users pick random private values **x (x < p) and y (y < p)**
- Compute public values
  - $R1 = g^x \bmod p$
  - $R2 = g^y \bmod p$
- Public values R1 and R2 are exchanged
- Compute shared, private key
  - $k_{alice} = (R2)^x \bmod p$
  - $k_{bob} = (R1)^y \bmod p$
- Algebraically it can be shown that $k_{alice} = k_{bob}$
  - Users now have a symmetric secret key to encrypt

Key Exchange



$K = g^{xy} \bmod p$

Proof

- We know

  $R1 = g^x \bmod p$

  $R2 = g^y \bmod p$

- $k_{alice} = (R2)^x \bmod p$

  $= (g^y \bmod p)^x \bmod p$

  $= (g^y)^x \bmod p$

  $= (g)^{yx} \bmod p$

  $= (g^x)^y \bmod p$

  $= (g^x \bmod p)^y \bmod p$

  $= (R1)^y \bmod p$

  $= k_{bob}$

Example

- Alice and Bob get public numbers
  - P = 19, G = 3 [Primitive roots of modulus 19 are 2,3,10,13,14,15]
- Alice and Bob pick private values x=15 & y=10 respectively
- Alice and Bob compute public values
  - R1 = 3^{15} mod 19 = 12
  - R2 = 3^{10} mod 19 = 16
  - Alice and Bob exchange public numbers
- Alice and Bob compute symmetric keys
  - kalice = $(R2)^x$ mod p = $(16)^{15}$ mod 19 = 7
  - kbob = $(R1)^y$ mod p = $(12)^{10}$ mod 19 = 7
- Alice and Bob now can talk securely!

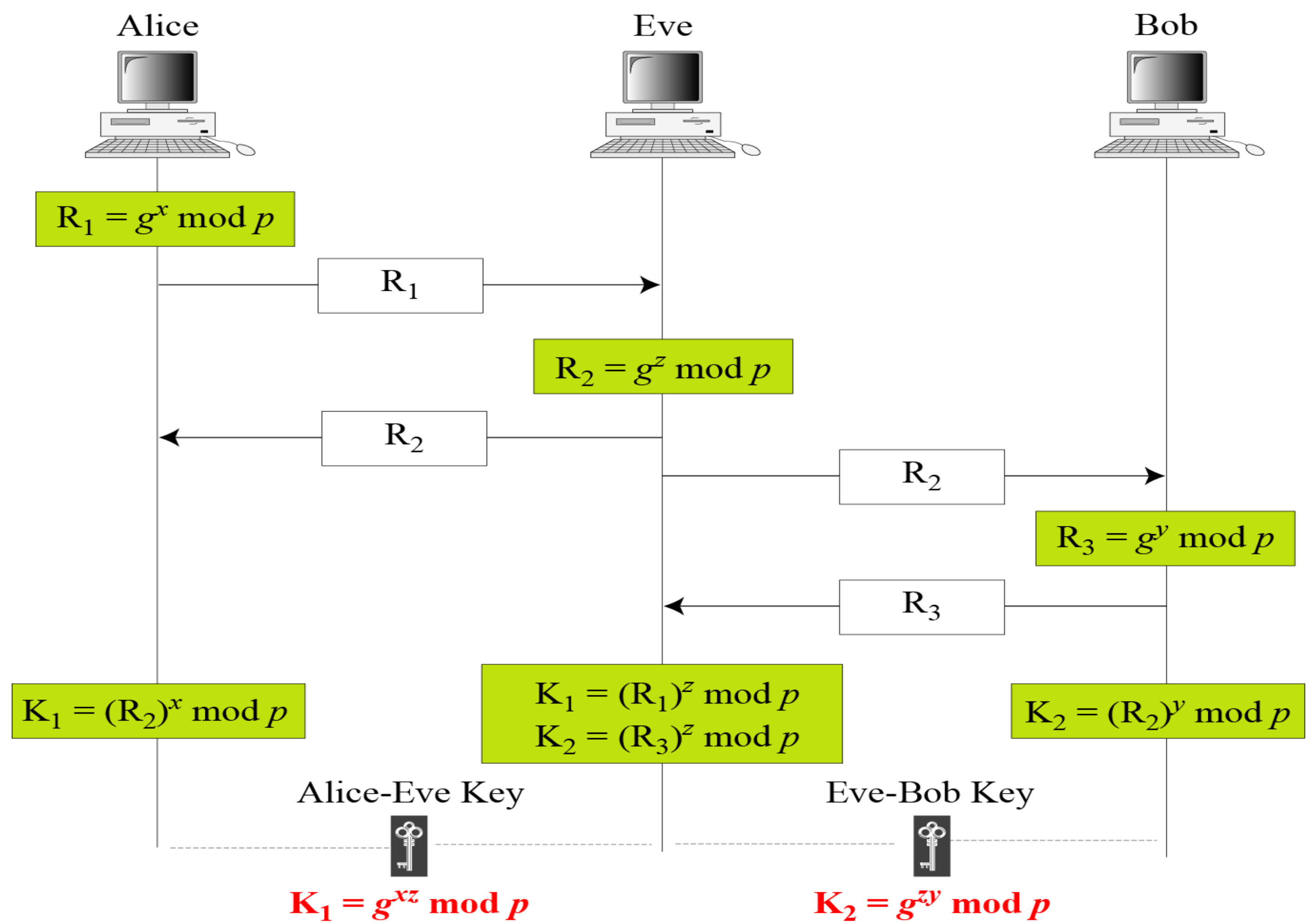
Security of Diffie-Hellman

- This protocol susceptible to two attacks:
  - The Man-in-the-middle attack
  - The Discrete logarithmic attack

Man-in-the-middle attack

- (p and g are publicly known)
- An adversary Eve intercepts Alice's public value and sends her own public value to Bob.
- When Bob transmits his public value, Eve substitutes it with her own and sends it to Alice.
- Eve and Alice thus agree on one shared key and Eve and Bob agree on another shared key.
- After this exchange, Eve simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party.
- This is present because Diffie-Hellman key exchange does not authenticate the participants.

Man-in-the-middle attack (cont.)



Example

- Alice and Bob get public numbers
  - P = 19, G = 3 [Primitive roots of modulus 19 are 2,3,10,13,14,15]
- Alice , Eve and Bob pick private values x=15 & z=12 & y=10 respectively
- Alice, Eve and Bob compute public values
  - R1 = 3^{15} mod 19 = 12
  - R2 = 3^{12} mod 19 = 11
  - R3 = 3^{10} mod 19 = 16
  - Alice , Eve and Eve , Bob exchange public numbers
- Alice and Bob compute symmetric keys
  - kalice = $(R2)^{x}$ mod p = $(11)^{15}$ mod 19 = 1
  - kbob = $(R2)^{y}$ mod p = $(11)^{10}$ mod 19 = 11

  - (K1) Eve = $(R1)^{z}$ mod p = $(12)^{12}$ mod 19 = 1
  - (K2) Eve = $(R3)^{z}$ mod p = $(16)^{12}$ mod 19 = 11

Discrete Logarithmic Attack

- For some values of the prime p, choose values of g such that g^x has only a small number of possible values, no matter what x is, which would make it easy to find for a value of x that was equivalent to the original value of x.
- Adversary knows p, g, R1, R2
- Adversary is forced to take discrete logarithm to determine key.
- $R1 = g^x \bmod p$
- $X = dlog_{gmodp}(R1)$

- E.g g=3, p=19, R1=8
- $dlog_{3mod19}(8) =$ ______
- So result = 3^3mod 19 =8 so x= 3

- Then Adversary can calculate K same as Bob calculates it.

Discrete Logarithmic Attack

- Consider a Diffie-Hellman scheme with a common prime p = 11 and a primitive root g = 2.
- If user A has public key R1 = 9, what is A's private key X?
- Adversary is forced to take discrete logarithm to determine key.
- $R1 = g^x \bmod p$
- $X = dlog_{gmodp}(R1)$

- E.g g=2, p=11, R1=9
- $dlog_{2mod11}(9) = \_\_\_\_\_\_$
- So result = 2^6mod 11 =9 so x= 6

- Then Adversary can calculate K same as Bob calculates it.

Discrete Logarithmic Attack (cont.)

- The security of this protocol depends on the discrete logarithm problem.
  - It assumes that it is computationally infeasible to calculate the shared secret key k = (g^{xy} mod p) given the two public values (g^x mod p) and (g^y mod p) when the prime p is sufficiently large (>= 1024 bits).

Discrete Logarithmic Attack

- For example, if *p*=170141183460469231731687303715884105727,
  - then it would take roughly 1.14824 µ 1021 steps to solve. (Each step requires many calculations.)
- Even using computers which are estimated to perform 300 trillion calculations per second, it would take roughly 5 years to solve.

Use of DH in Secure Internet Protocols

- DH in SSL
  - Secure Sockets Layer (SSL) is a de-facto standard for securing information flow between web users and web servers.
  - In SSL, the Handshake Protocol is responsible for authentication of the parties and negotiation of encryption methods and keys.
  - In this process, DH can be used. DH is considered the strongest alternative of the available options for the key exchange

Use of DH in Secure Internet Protocols

- DH in SSH
  - Secure Shell (SSH) is a both a protocol and a program used to encrypt traffic between two computers.
  - The two parties to the connection (e.g., client and server) begin their conversation by negotiating parameters (e.g., preferred encryption and compression algorithms, and certain random numbers).
  - Then a shared secret is computed using DH

Summary

- Key agreement protocol
- Strength based on discrete logarithms
- Does:
  - Allow public key exchange with no prior shared secrets
  - Works very efficiently
- Does not
  - Provide encryption, decryption, signature, etc.
  - Authenticate

- Defeats Man-in-the-middle attack
- Defeats Discrete logarithm attack