

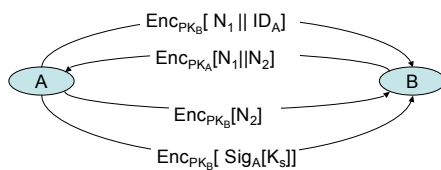
Key distribution and agreement

via public key cryptography

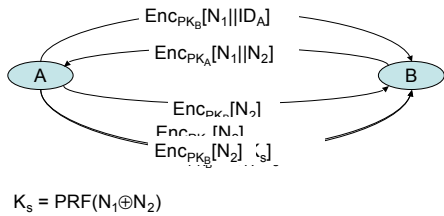
Authenticated key distribution

- Two parties, A and B , have public keys PK_A and PK_B , respectively.
- They may use the public keys to encrypt secret keys to each other, respectively.
- Unless the public keys are distributed in a *trustworthy* manner the exchange is vulnerable to *impersonation attacks*.

Needham-Schroeder key distribution



Criticizing Needham-Schroeder



Key agreement

- Key distribution is a particular case of key agreement.
 - In key agreement, two parties “negotiate a key,” obtained through joint generation
- Key agreement usually has secondary goals, such as forward security:
 - A key agreement protocol that is forward secure results in the session keys remaining private even if the private keys of the parties are compromised.

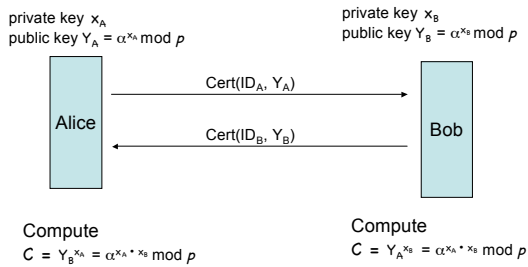
Discrete log-based Public Keys

- Let $Z_p^* = \{1, 2, \dots, p-1\}$.
- A primitive root α of Z_p^* is one such that $\{\alpha, \alpha^2, \alpha^3, \dots, \alpha^{p-1} = 1\} = Z_p^*$. (All operations mod p).
- Let p and α be common system parameters (known to all).
- Choose private key x in $\{1, \dots, p-1\}$ and compute the public key as $Y = \alpha^x$.

The discrete logarithm problem (DLP)

- Given elements α and Y in Z_p^* , find x such that $Y = \alpha^x \bmod p$ (provided that such x exists) is the discrete logarithm problem (modulo p).
- This is a long-standing problem and no polynomial algorithms are known for general primes.
 - Therefore, the private key of the previously described public key scheme remains secret

Diffie-Hellman Key Exchange



Security of DH Key-Exchange

- An eavesdropper will recover A and B .
 - If the eavesdropper could recover either x_A or x_B , it would be able to compute the shared secret.
- **The discrete logarithm problem (DLP):**
Given α , and A , find x such that :
 - $A = \alpha^x \bmod p$ (assume α a generator)
- For DHKE to be secure, it is required that there not exist an efficient algorithm to solve DLP.

The Diffie-Hellman Problem

- To compute the shared secret, all the eavesdropper needs to learn is how to solve the
- **Diffie-Hellman problem (DHP):**
- **Given A and B, find C such that**
 - $A = \alpha^r$; $B = \alpha^s$; and $C = \alpha^{rs}$ (all mod p),
 - α a generator.
- Not required to learn r or s .

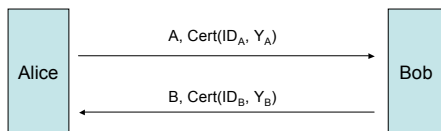
Summary -- Security of DHKE

- The security of Diffie-Hellman Key exchange depends on the hardness of the Diffie-Hellman problem (DHP).
- If there exists an efficient algorithm to compute discrete logarithms, the DHP can be solved as well. So the security of DHKE also depends on the hardness of the DLP (indirectly).

Randomized DHKE

private key x_A ,
public key $Y_A = \alpha^{x_A} \text{ mod } p$
generate random r
compute $A = \alpha^r \text{ mod } p$

private key x_B ,
public key $Y_B = \alpha^{x_B} \text{ mod } p$
generate random s
compute $B = \alpha^s \text{ mod } p$



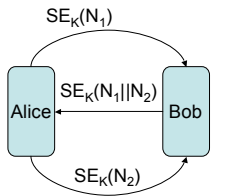
Compute $K = B^{x_A} Y_B^r \text{ mod } p$
 $= \alpha^{x_A s + x_B r} \text{ mod } p$

Compute $K = A^{x_B} Y_A^s \text{ mod } p$
 $= \alpha^{x_B r + x_A s} \text{ mod } p$

Properties of randomized DHKE

- Requires a follow-up by a 3-way handshake to guarantee liveness of the other party (next)
- Provides forward security: Even if one of the private keys is compromised, an eavesdropper cannot recover old session keys from recorded transactions (because does not know the randomized values used).

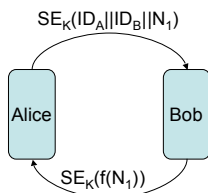
3-way handshake



Analogous to a public key protocol (fig. 10.3), messages 3, 6, 7.

- Guarantees that the interchange took place with the intended party, and a key was indeed established.
 - Absent this, DHKE could take place with a different party (who would not recover the secret but would lead one of the parties to believe the other had it.)

2-way handshake



- The 3-way handshake (needed in the public-key case) can be reduced to 2-way in symmetric key case.
 - Since key K just computed, liveness of A is guaranteed after 1st message
