Diffie-Hellman key exchange

Key Exchange

- We have seen already how public-key cryptography may be used for public key distribution;
- Public-key cryptography may be used also for key exchange:
  - Two parties (users) execute some algorithm (protocol) and get a common secret key;
  - The key may be used for subsequent encryption of messages;

Diffie-Hellman Key Exchange

- Most known algorithm for key exchange is Diffie-Hellman algorithm (1976);
- The purpose of the algorithm is exchange of a secret key (not encryption);
- DH algorithm is considered as a public-key algorithm because:
  - Users to generate the same secret key rely on publicly known information + some private information;
  - In principle, it is possible to generate a key knowing only public information, but it is computationally expensive;

Discrete logarithms

- Security of DH algorithm relies upon difficulty of computing *discrete logarithms;*
- *Primitive root of a prime number p:* a number a such that all numbers

$a \bmod p, a^2 \bmod p, \ldots, a^{p-1} \bmod p$ are different;

- For any number *b* less than *p* and a primitive root a of *p* the discrete logarithm (index) of *b* for the base a *mod p* is the number i *such that*

$$b = a^i \bmod p \qquad 0 \le i \le (p-1)$$

Discrete logarithms

- Notation: $ind_{a,p}(b)$
- Key facts:
  - It is relatively easy calculate exponentials modulo a prime, that is given a,*i,p* calculate $a^i \bmod p$
  - It is very difficult and for large primes infeasible to calculate discrete algorithms, that is given *b,a,p* find i such that

$$b = a^i \bmod p$$

Diffie-Hellman key exchange

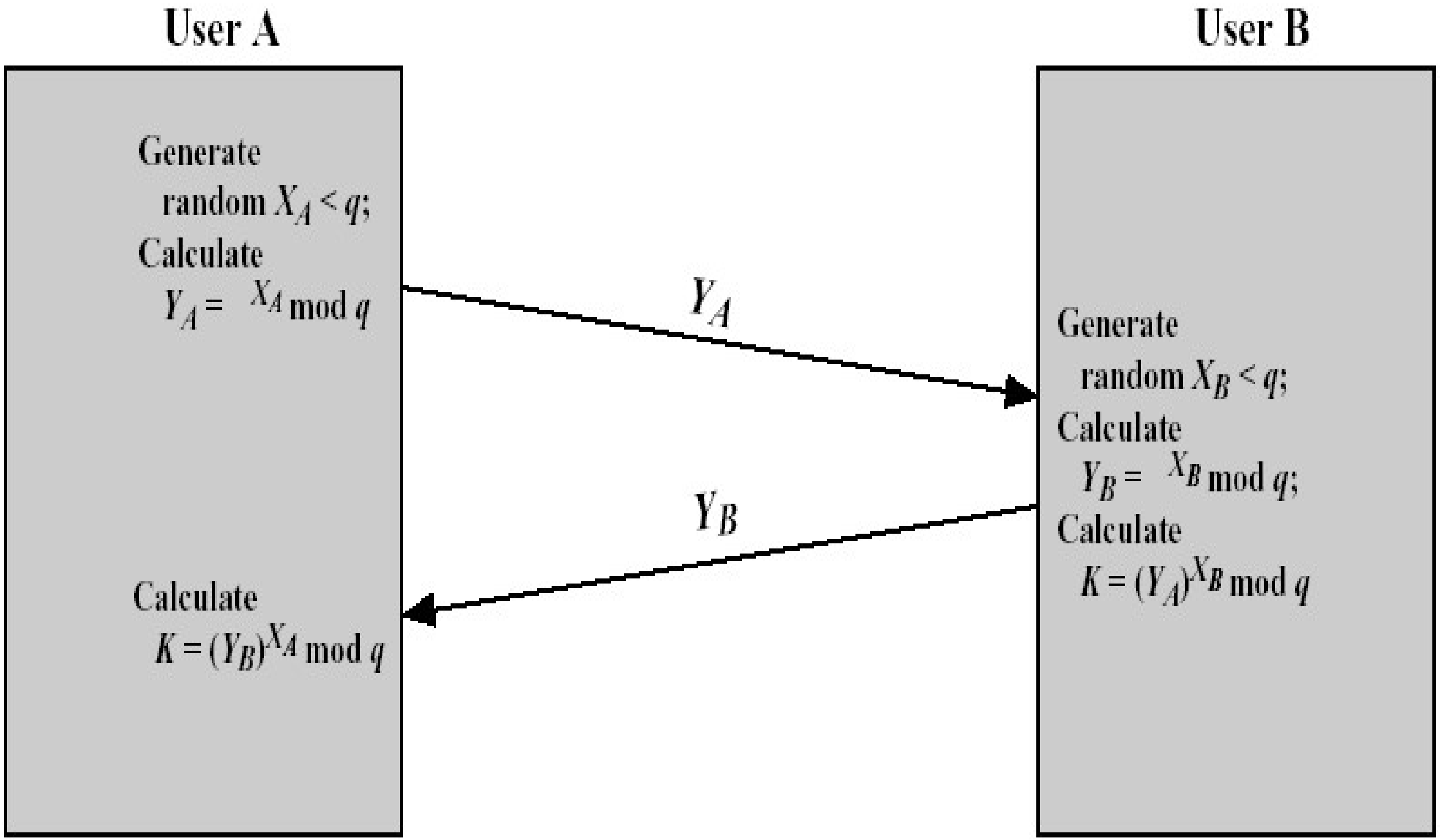
- Two publicly known numbers:
  - prime number *q*
  - primitive root α of *q*
- Let *A* and *B* wish to exchange a key, then they do the following:
  - *A* selects a random integer $X_A < q$ and keeps it in secret
  - *B* selects a random integer $X_B < q$ and keeps it in secret
  - *A* computes $Y_A = \alpha^{X_A} \bmod q$ and sends it to *B*
  - *B* computes $Y_B = \alpha^{X_B} \bmod q$ and sends it to *A*

The secret key

- Both A and B is now able to calculate common secret key:
  - A calculates $K = (Y_B)^{X_A} \bmod q$
  - B calculates $K = (Y_A)^{X_B} \bmod q$

- These calculations give identical results and *K* is the common secret key.

Deffie-Hellman Key Exchange



How to break HD key exchange?

- An attacker knows q, a, Y_A, Y_B
- How can (s)he calculate *K?*
- Straightforward way is to find out X_A, or X_B and repeat calculations of *A* or *B*;
- However this includes calculations of discrete logarithms: $X_B = ind_{\alpha,q}(Y_B)$ which is infeasible for large *q;*
- No essentially better passive attacks are known.

Example

- For q = 7 check that 2 **is not** a primitive root of 7 and 3 **is** a primitive root of 7;
- Let *q = 7* and *a = 3* is publicly known numbers in DH algorithm;
- Let $X_A = 4$ and $X_B = 3$ be private keys of A and B, respectively;
- Then $Y_A = 3^4 \bmod 7 = 4$
- $Y_B = 3^3 \bmod 7 = 6$
- Common secret key

$$K = 6^4 \bmod 7 = 4^3 \bmod 7 = 1$$