Lecture 27

Diffie-Hellman Key Exchange

Section 3.5

Diffie-Hellman Key Exchange

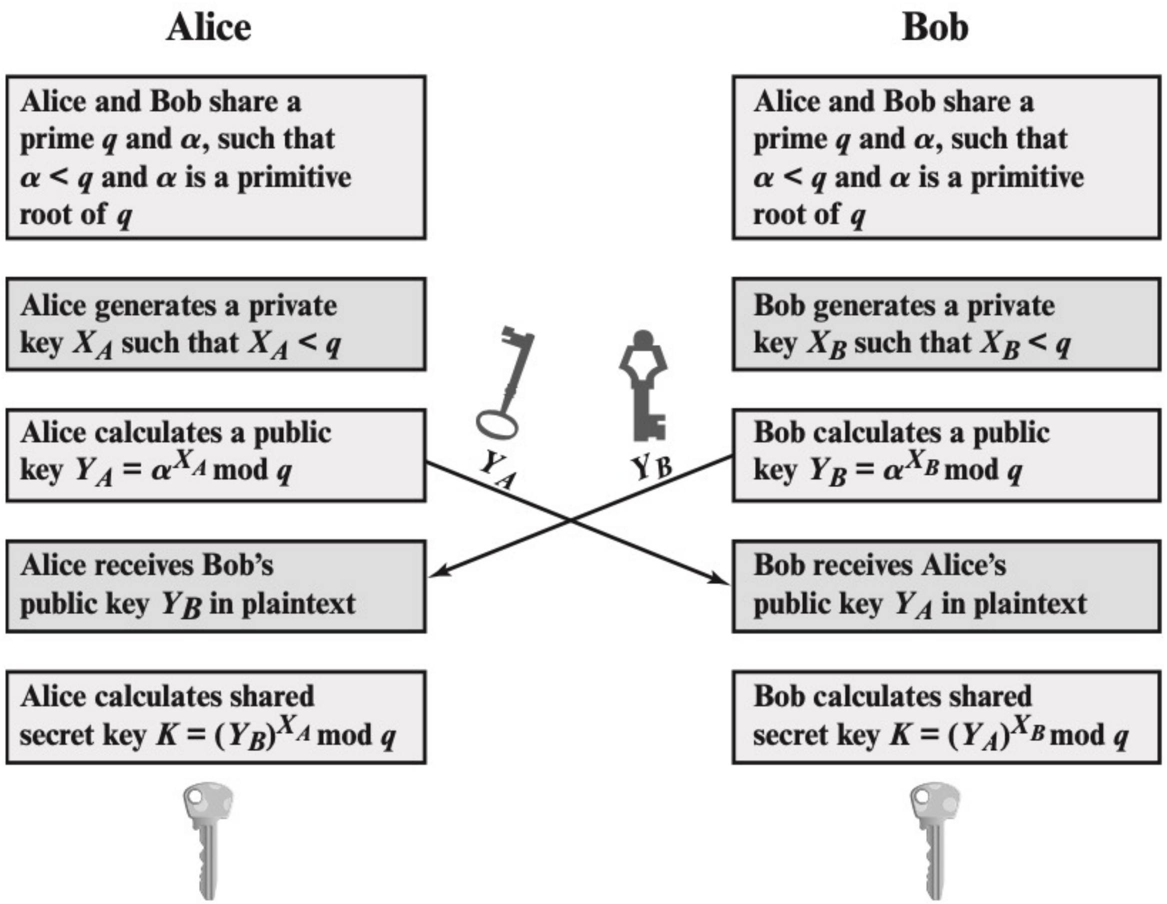
- No third party involved
- After a common shared key, $\alpha^{X_A X_B}$ is established, it can be used to encrypt message
- A common shared key is symmetric

The Diffie-Hellman Key Exchange

- From B's view
- $K = {Y_B}^{X_A} \, mod \; q$

  $= (\alpha^{X_B} \, mod \; q)^{X_A} \, mod \; q$

  $= \alpha^{X_B X_A} \, mod \; q$



Example

- $given\ q = 353, \alpha = 3, X_A = 97, X_B = 233$
- A computes $Y_A = 3^{97}\ mod\ 353 = 40.$ B computes $Y_B = 3^{233}\ mod\ 353 = 248$
- Then communication key exchange - Y_A, Y_B
- A receives Y_B. B receives Y_A
- A computes $K = {Y_B}^{X_A}\ mod\ 353 = 248^{97}\ mod\ 353 = 160$

  B computes $K = {Y_A}^{X_B}\ mod\ 353 = 40^{233}\ mod\ 353 = 160$

Attack

- Adversary gets q, α, Y_A, Y_B.
- She needs to compute either X_A or $X_B = dlog_{\alpha,p} Y_B$
- Secure?

Discrete Log Problem

Two cryptographic assumptions:

- **Discrete logarithm problem** (**discrete log problem**): Given $\alpha, q, \alpha^{X_A} \, mod \, q$ for random X_A, it is computationally hard to find X_A
- **Diffie-Hellman assumption**: Given $\alpha, q, \alpha^{X_A} \, mod \, q$, and $\alpha^{X_B} \, mod \, q$ for random X_A, X_B, no polynomial time attacker can distinguish between a random value R and $\alpha^{X_A X_B} \, mod \, q$.
  - Intuition: The best known algorithm is to first calculate X_A and then compute $(\alpha^{X_B})^{X_A} \, mod \, q$, but this requires solving the discrete log problem, which is hard!
- Note: Multiplying the values doesn't work, since you get $\alpha^{X_A + X_B} \, mod \, p \;\neq\; \alpha^{X_A X_B} \, mod \, p$