Diffie-Hellman Key Exchange

Section 3.5

Outline

- Diffie-Hellman Key Exchange (DHKE) Algorithm
- Analysis of DHKE

Use case

Attack
- Passive: Eve. no modification of msg
- Active: modify msg

(1) Sniffer.
(2) ISP
(3) Government

Alice

Bob

msg $\xrightarrow{\text{unsecure}}$

Encryption

(1) Symmetric → pre-shared secret key

(2) A~~sym~~metric

How to distribute the symmetric key?

4 ways:
1. physical deliver
2. third authority physical deliver
3. old key encrypts new
4. A $\xrightarrow{\text{Encrypted}}$ C $\xrightarrow{\text{Encrypted}}$ B

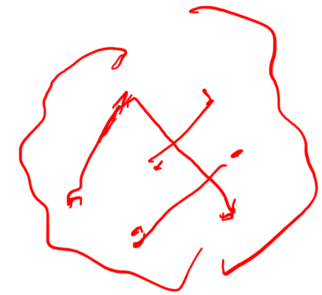
Recall: ways to achieve symmetric key distribution

geographically far

- A key could be selected by A and physically delivered to B
- A third party could select the key and physically deliver it to A and B
- If A and B have previously and recently used a key, one party could transmit the new key to the other, using the old key to encrypt the new key   How to distribute old key
- If A and B each have an encrypted connection to a third-party C, C could deliver a key on the encrypted links to A and B

No third authority
create and distribute key service

Diffie-Hellman Key Exchange



- Invented by Whitfield Diffie and Martin Hellman in 1976
- Allows Alice and Bob to exchange a key without Eve learning it
- No third party involved
- After DHKE, 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

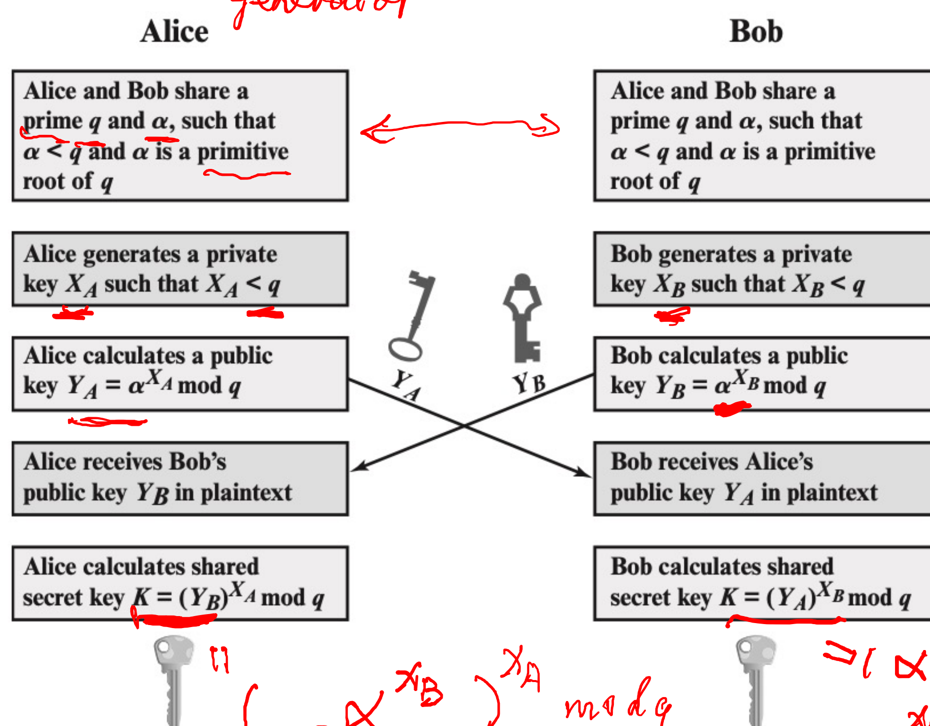
$\{\alpha^{x} \bmod q \mid x = 1, \ldots, n\} \in [1, q-1]$

generator

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

  $= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$

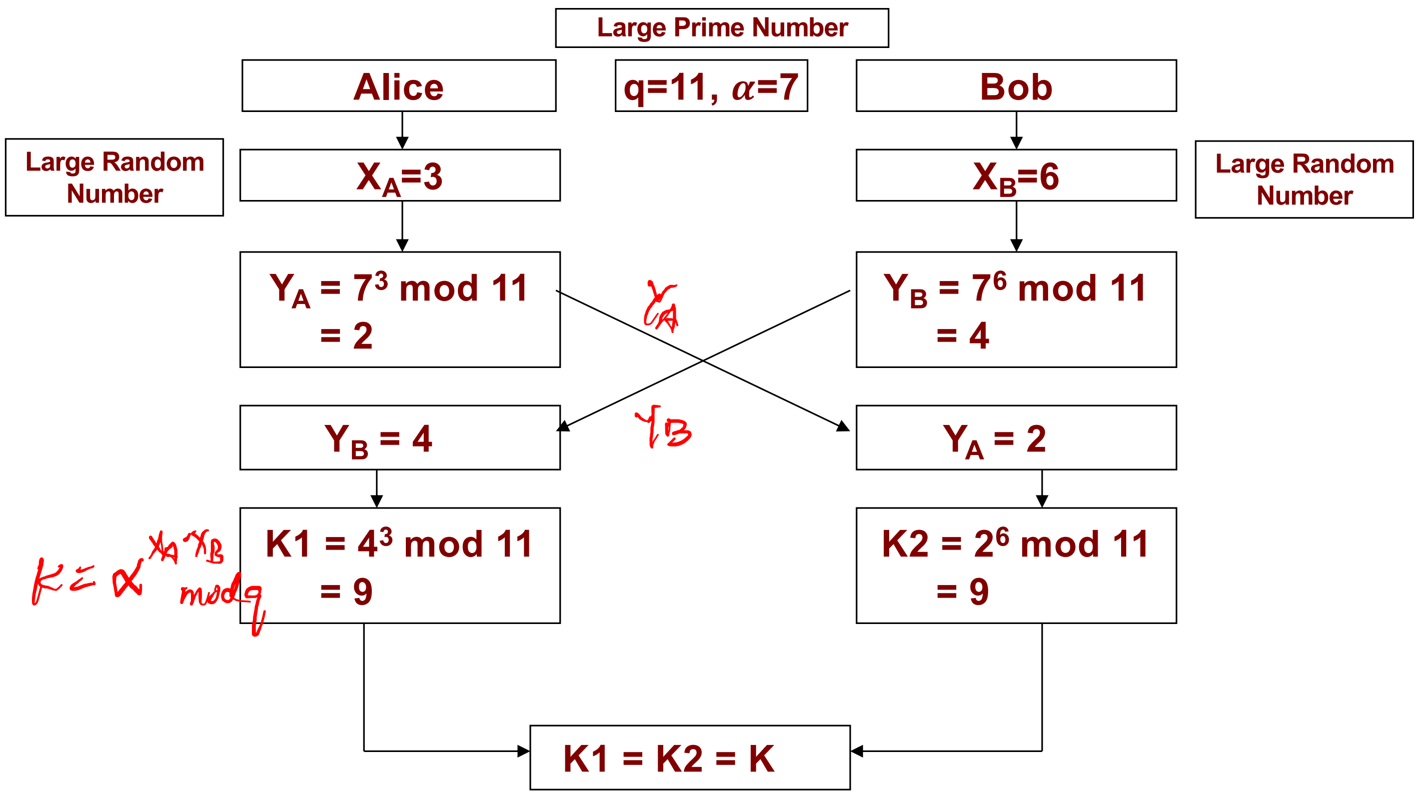
  $= \alpha^{X_B X_A} \bmod q$



$= (\alpha^{X_B})^{X_A} \bmod q = \alpha^{X_B \cdot X_A} \bmod q$

$= (\alpha^{X_A})^{X_B} \bmod q = \alpha^{X_A \cdot X_B} \bmod q$

Example of the Diffie -Hellman algorithm



Note: X_A, X_B, K1, K2 are Private to others

Analysis of DHKE - Attack

$Y_A = \alpha^{X_A}$ $\quad X_A = dlog_{\alpha,q} Y_A \bmod q$

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

Discrete Log Problem

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 $\alpha^{X_A+X_B}\ mod\ p\ \neq\ \alpha^{X_A X_B}\ mod\ p$

DHKE in Python Cryptography Library

- https://cryptography.io/en/latest/hazmat/primitives/asymmetric/

Summing Up

- Symmetric Key crypto has a major problem:
  - How do two people who don’t know each other share a key?
- A Diffie-Hellman key exchange lets them compute a shared key even in the presence of an eavesdropper, Eve.
- However, if attack is active, instead of passive, this wouldn’t work …
- Diffie-Hellman suffers man-in-the-middle attack (next class)

Take home exercises

- SW, "Network Security Essentials", 6th Edition, 2017
  - Problems – 3.21

  Consider a Diffie-Hellman scheme with a common prime q = 11 and a primitive root α = 2.

    a. if user A has public key Y_A = 9, what is A's private key X_A?

    b. If user B has public key Y_B = 3, what is the shared secret key K?