Fundamentals of Information & Network Security
ECE 471/571

Lecture #27: Key Distribution & Management

Instructor: Ming Li

Dept of Electrical and Computer Engineering

University of Arizona

The Key Distribution Problem

- Key Pre-Distribution
  - A TA (Trusted Authority) distributes keying info ahead of time via secure channel to every user
  - Every pairs of users will be able to determine a shared key (non-interactively)
- Session Key Distribution
  - TA chooses session key and distributes to everyone via an interactive protocol
  - Session keys are encrypted by the pre-distributed keys
  - Session keys are used to encrypt for a fairly short period of time
- Key Agreement
  - Two users can generate a shared session key interactively without an online TA

Long-Lived Keys and Session Keys

- Long-lived keys are pre-computed and stored securely
  - Use long-term shared keys (public/private keys or secret keys) to authenticate.
- Authentication protocols negotiate session keys for subsequent data encryption
  - Session key is usually secret key, more efficient for encryption

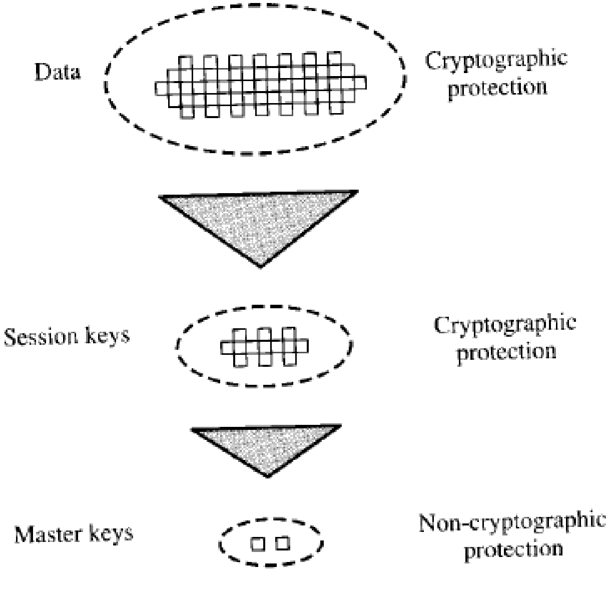


Figure 7.8 The Use of a Key Hierarchy

- Why session keys?
  - Limit the amount of ciphertext available to the attacker (Keys sort of “wear out” if used a lot)
  - Limit the risk of exposure of the long-term key during compromise
  - Reduce the amount of secret information storage needed
  - Shared key encryption is subject to replay attacks.

Key Distribution

Symmetric key problem:

- How do two entities establish shared secret key over network?

Solution:

- trusted key distribution center (KDC) acting as intermediary between entities

Public key problem:

- When Alice obtains Bob’s public key (from web site, e-mail, diskette), how does she know it is Bob’s public key, not Trudy’s?

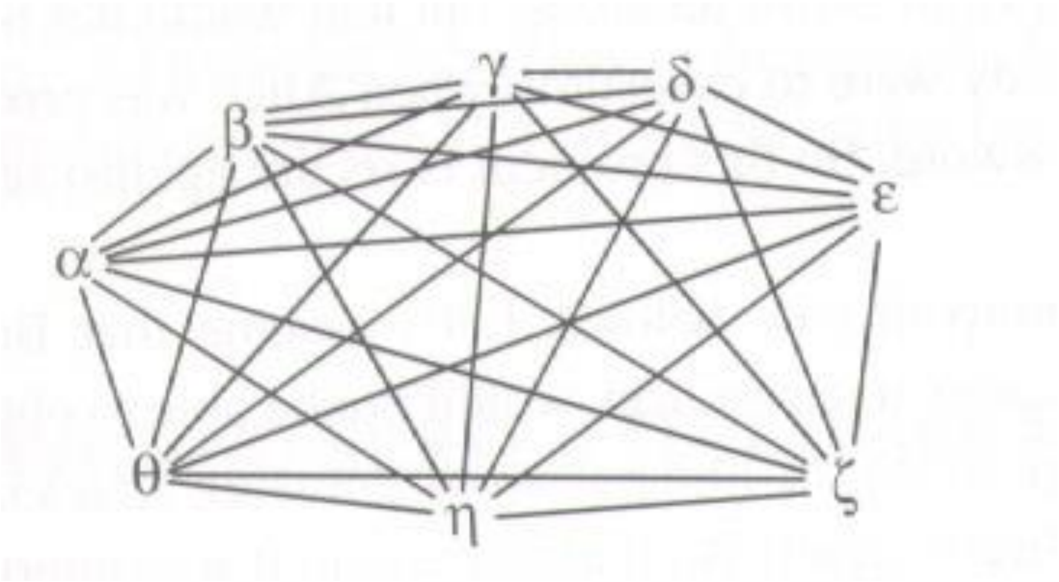
Solution:

- trusted certification authority (CA)

Trusted Intermediaries

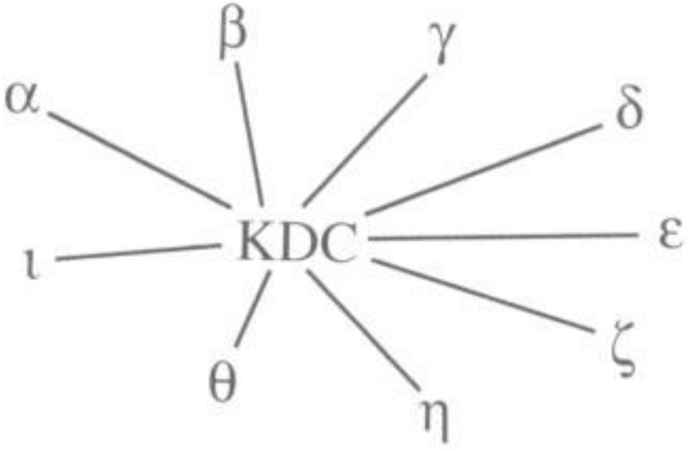
Scalability Problem of Symmetric Keys

- For *n* machines to mutually authenticate each other, O(n^2) number of keys are required.



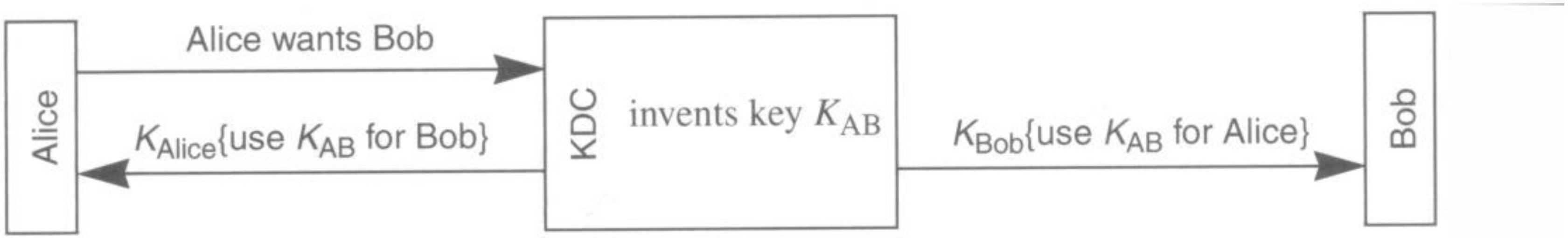
Solution: KDC

- Key Distribution Center
  - a trusted node
  - Each node, *i*, has a secret key with the KDC, K_i



Authentication with KDC (in Principle)

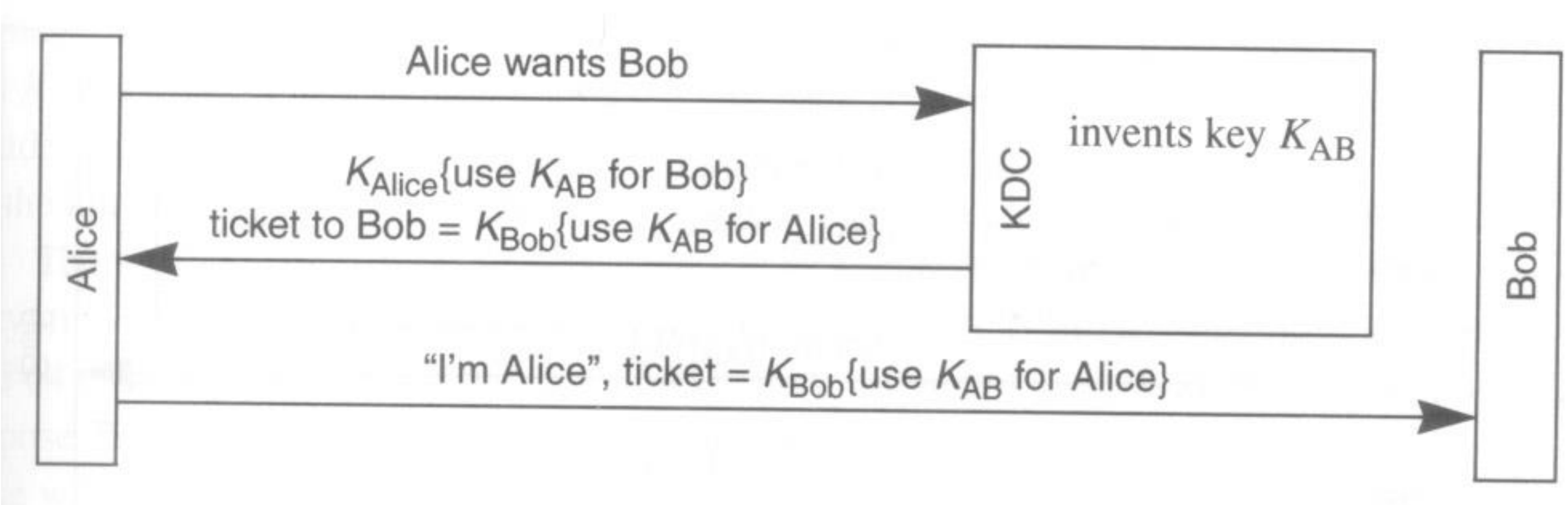
- A and B do not authenticate directly.
- A first authenticates with KDC;
- KDC selects a temporary secret K_{AB}, and sends $E_{K_A}\{K_{AB}\}$ to A and $E_{K_B}\{K_{AB}\}$ to B.
- Now A and B has a common secret K_{AB}, and they can authenticate each other.


Alice
Alice wants Bob
K_{Alice}\{use K_{AB} for Bob\}
KDC
invents key K_{AB}
K_{Bob}\{use K_{AB} for Alice\}
Bob

Protocol 11-16. KDC operation (in principle)

Authentication with KDC (in Practice)

- A and B do not authenticate directly.
- A first authenticates with KDC;
- KDC selects a temporary secret K_{AB}, and sends $E_{K_A}\{K_{AB}\}$ and $E_{K_B}\{K_{AB}\}$ to A.
- A sends $E_{K_B}\{K_{AB}\}$, called a ticket, to B.
- Now A and B has a common secret K_{AB}, and they can authenticate each other.


Alice
Alice wants Bob
KDC
invents key K_{AB}
K_{Alice}\{use K_{AB} for Bob\}
ticket to Bob = K_{Bob}\{use K_{AB} for Alice\}
"I'm Alice", ticket = K_{Bob}\{use K_{AB} for Alice\}
Bob

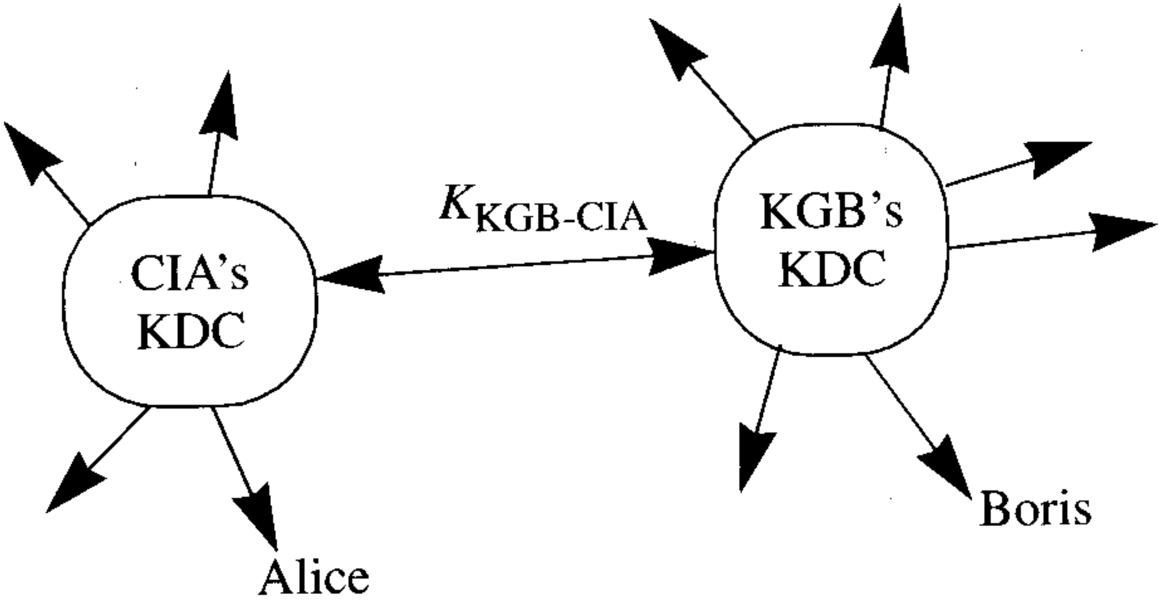
Protocol 11-17. KDC operation (in practice)

KDC Scalability

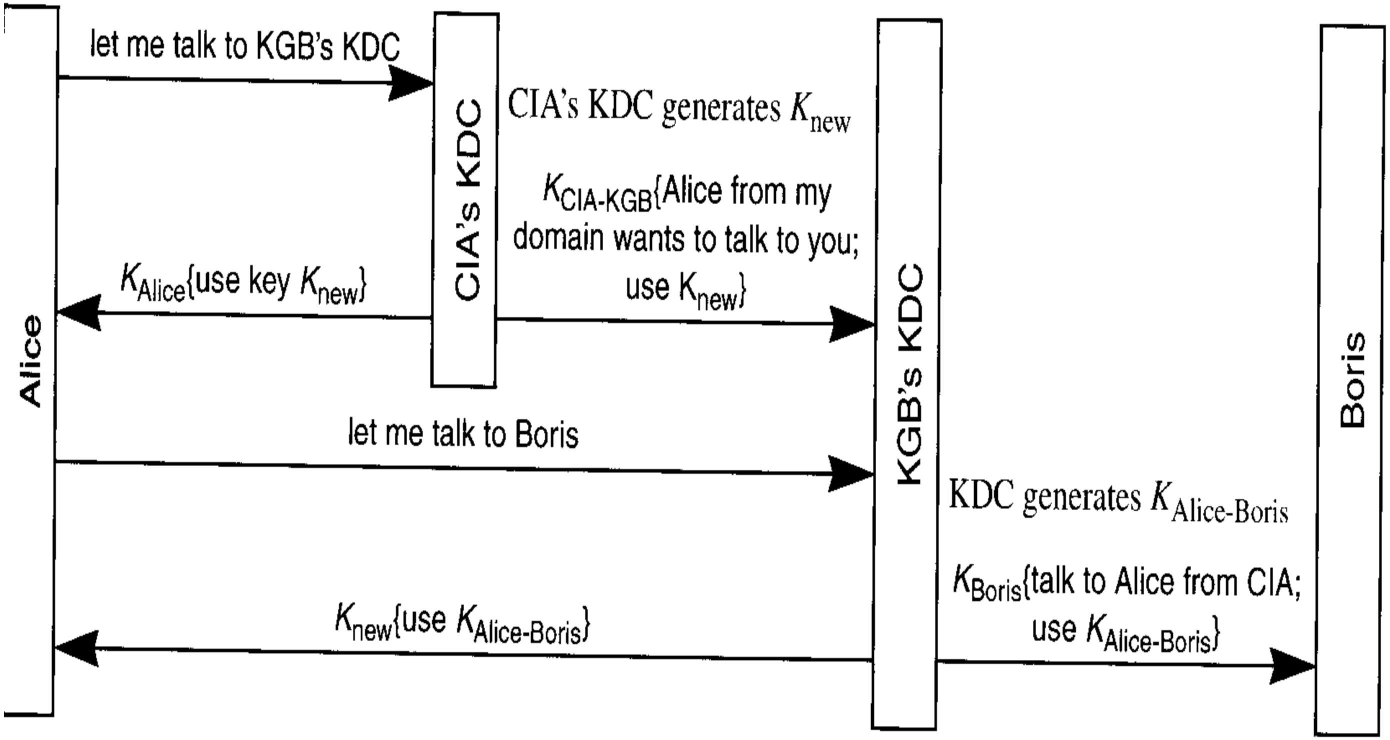
- O(n) keys are needed.
- When a new user arrives or a user key is compromised, only one place (KDC) and one key needs to be re-configured.
- Disadvantages
  - Single point of vulnerability
  - Single point of failure
  - Performance bottleneck

- Solution?

Multiple KDC Domains

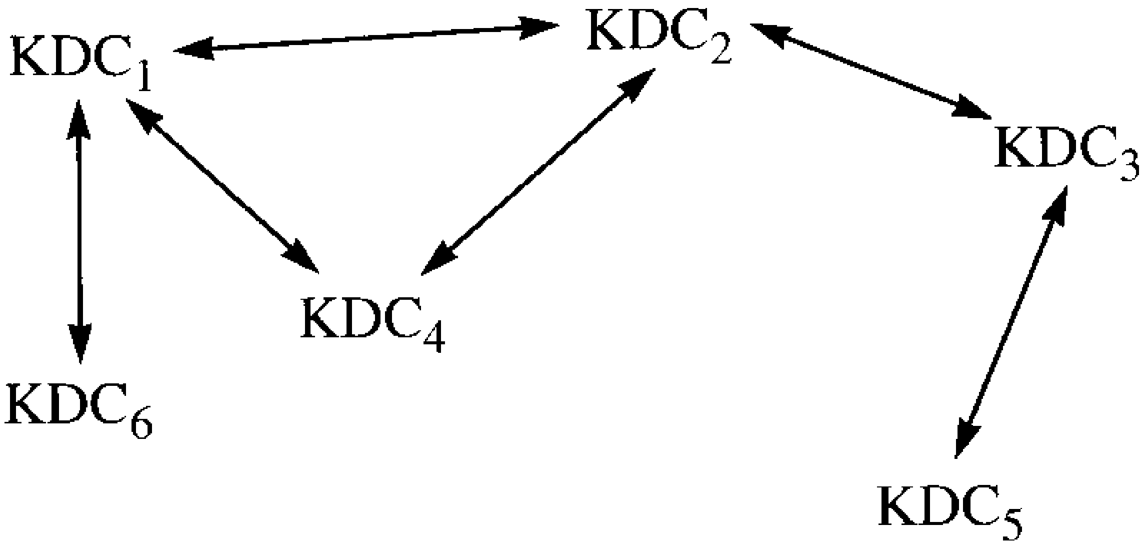


Authentication Across Domains

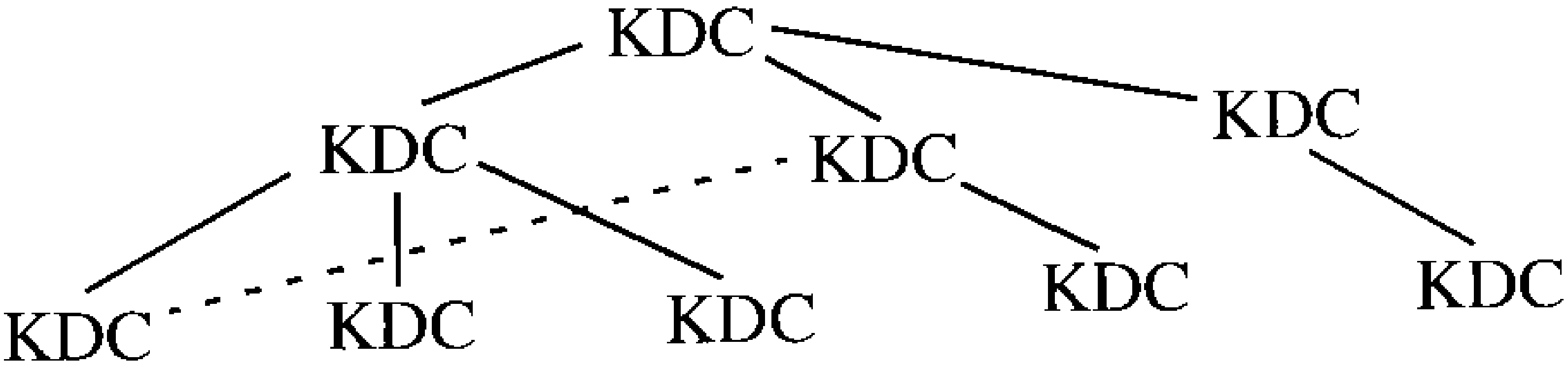


Alice
CIA's KDC
KGB's KDC
Boris
let me talk to KGB's KDC
CIA's KDC generates K_{new}
$K_{CIA\text{-}KGB}$\{Alice from my domain wants to talk to you; use K_{new}\}
K_{Alice}\{use key K_{new}\}
let me talk to Boris
KDC generates $K_{Alice\text{-}Boris}$
K_{Boris}\{talk to Alice from CIA; use $K_{Alice\text{-}Boris}$\}
K_{new}\{use $K_{Alice\text{-}Boris}$\}

Authentication by KDC Chains



KDC Hierarchy


KDC
KDC
KDC
KDC
KDC
KDC
KDC
KDC
KDC
KDC