Kerberos (V4)

Introduction

• Kerberos is a secret key based authentication service
• Based on work by Needham and Schroeder
• First three versions no longer in use
• V4 and V5 are competing for market
  – V4 has a greater installation base, simpler, and performs better
  – V5 has enhanced functionalities
• We study V4
  – Refer to the text book for V5 to satiate your curiosity
User Model

- A login session with multiple remote resource access sessions during the login session

Key Distribution Center

- Kerberos relies on a trusted key distribution center (KDC)
- At different context, also called ticket-granting server (TGS) or authentication server (AS)
  - No real distinction
- KDC shares a secret key with each principal
  - Also known as the master key
  - Stored in a database
Main Idea of Kerberos

• When a user logs in, he will receive a session key and a ticket-granting ticket
  – The latter is called TGT
• Whenever the user needs access to some resource, his session key and TGT can help him to obtain a ticket for using that service

Obtaining a Session Key and TGT

When Alice logs in

Alice, password

[AS_REQ]
Alice needs a TGT

[AS_REP]

Kₐ{Sₐ, TGT}

KDC

Invents key Sₐ
Finds Alice’s master key Kₐ
TGT = Kₐ{“Alice”, Sₐ}
Getting a Ticket to Bob for Alice

Alice: Hi, KDC, I am Alice, I need Bob’s service . . .

Logging into Bob from Alice’s Workstation

Alice: Bob, I need your service. Here is my ticket!
Problems with a Single KDC

• Single point of failure
• Performance Bottleneck

• Solution: Replicated KDCs

Replicated KDCs

• Each KDC must be interchangeable with every other KDC
• They share the same $K_{KDC}$
• They have the same identical databases of principal names and master keys
  – One site to keep the master copy
    • Any updates must be made here
  – Other sites periodically synchronize their copies
• Question: what if the master is down?
Can Everybody Trust a Single KDC (or multiple replicated ones)?

- The question can be rephrased as: can a single principal master key database work?
- A big network can have thousands of organizations and millions of users
- A KDC that everybody trusts seems unreasonable!
  - Remember that a KDC manages every registered principal’s master key!

Realms

- Principals are divided into realms
- Each realm has its own KDC database
- There can be multiple replicated KDCs in the same realm
Inter-Realm Authentication

- Assume two realms: Realm Wonderland and Realm X
- If Realm X is willing to provide services to principals in Realm Wonderland, the KDC for X registers can be registered as a principal in realm Wonderland

Diagram:

- Alice
  - TGS_REG("Alice@Wonderland", "X@Wonderland")
  - Credentials to X
  - TGS_REG("Alice@Wonderland", "Bob@X")
  - Credentials to Bob
  - AP_REQ

- Wonderland KDC
- X KDC
- Bob
Quiz 2

• Write what’s the contents for Alice’s credential to X and the credential to Bob