



**Cerberus** (from *Kerberos, demon of the pit*): Monstrous three-headed dog (sometimes said to have fifty or one-hundred heads), (sometimes) with a snake for a tail and innumerable snake heads on his back.

He guarded the gate to Hades (the Greek underworld) and ensured that the dead could not leave and the living could not enter.

# Kerberos (v.4)

- Kerberos service has one trusted server.
  - This server authenticates *principals* and implements an access control policy.
  - Each principal has a password-derived *master key*.
  - *Tickets* are encrypted session keys for use between a pair of principals.

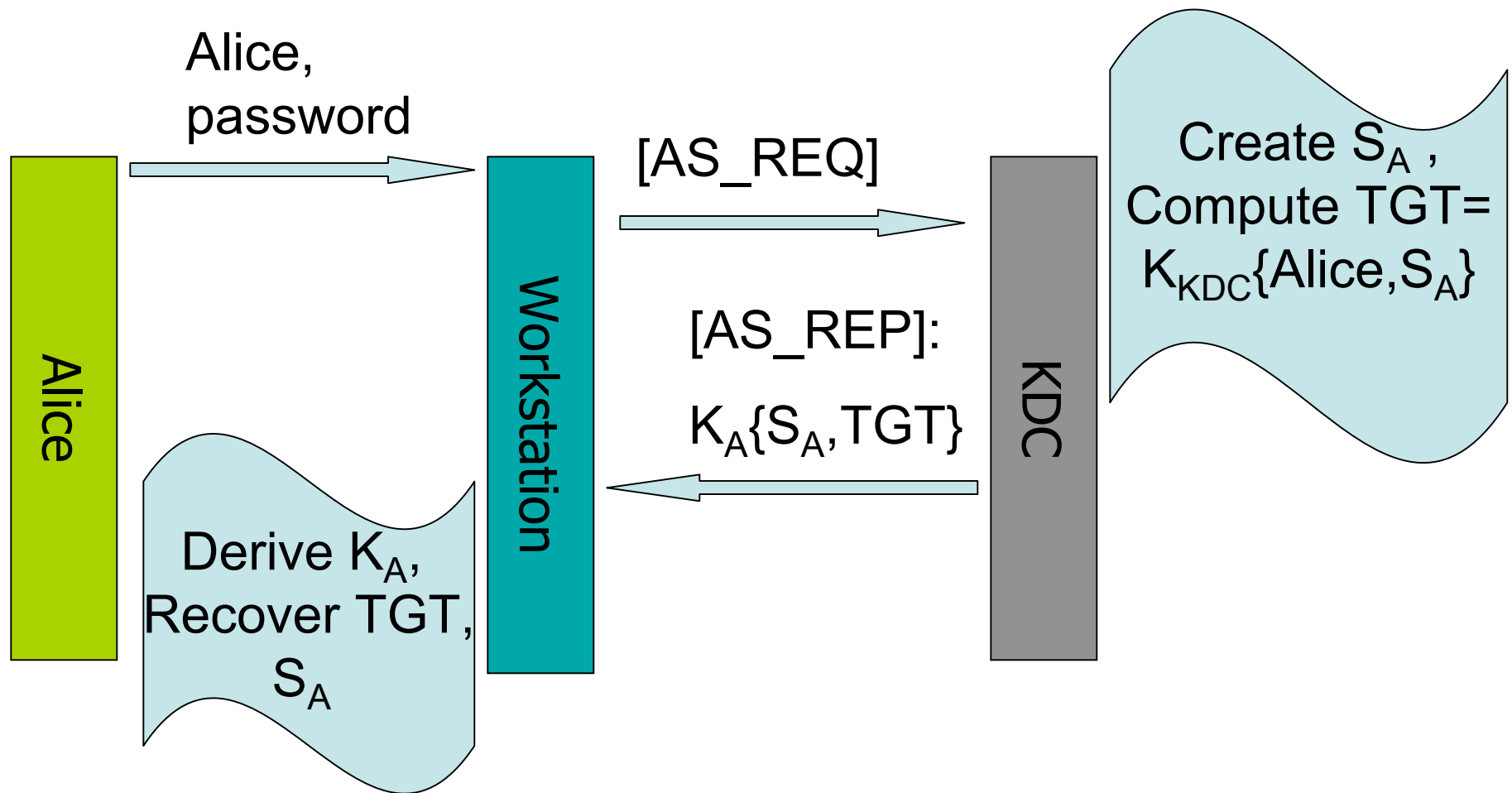
# The KDC

- The trusted server is referred to as:
  - AS (authentication server):
    - Authenticates users based on their master keys, hands off session keys for secondary authentication: *ticket-granting tickets* (TGT).
  - TGS (ticket granting server):
    - Performs secondary authentication based on the TGT keys, hands off tickets for principals to communicate with *kerberized* network services.

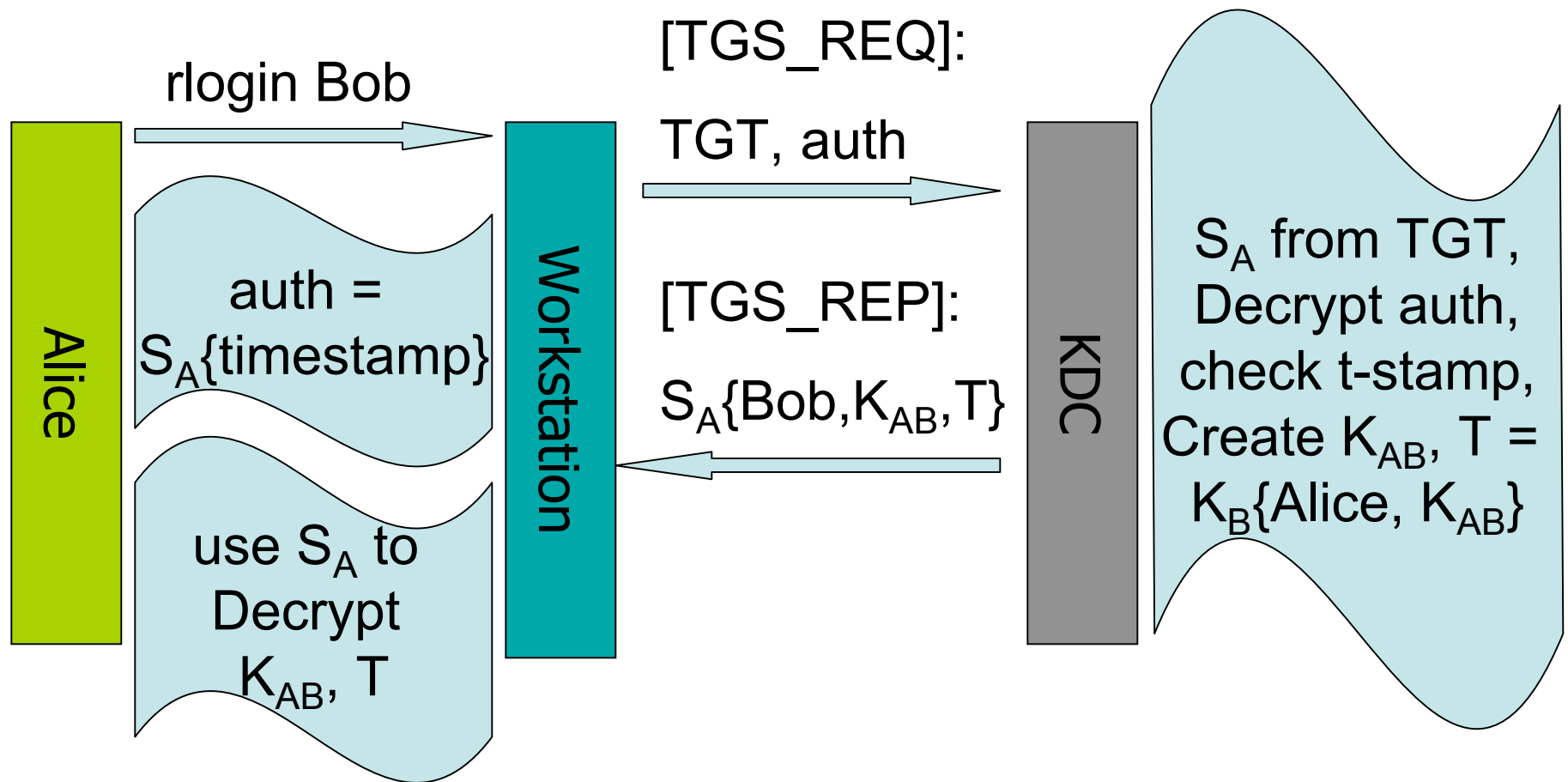
# KDC configuration

- The KDC database is kept encrypted under KDC's own master key.
- Users have passwords, their master key is derived from them
- Kerberized network servers need to store their master key somehow.
- All master keys are stored in the KDC database (except KDC's own.)

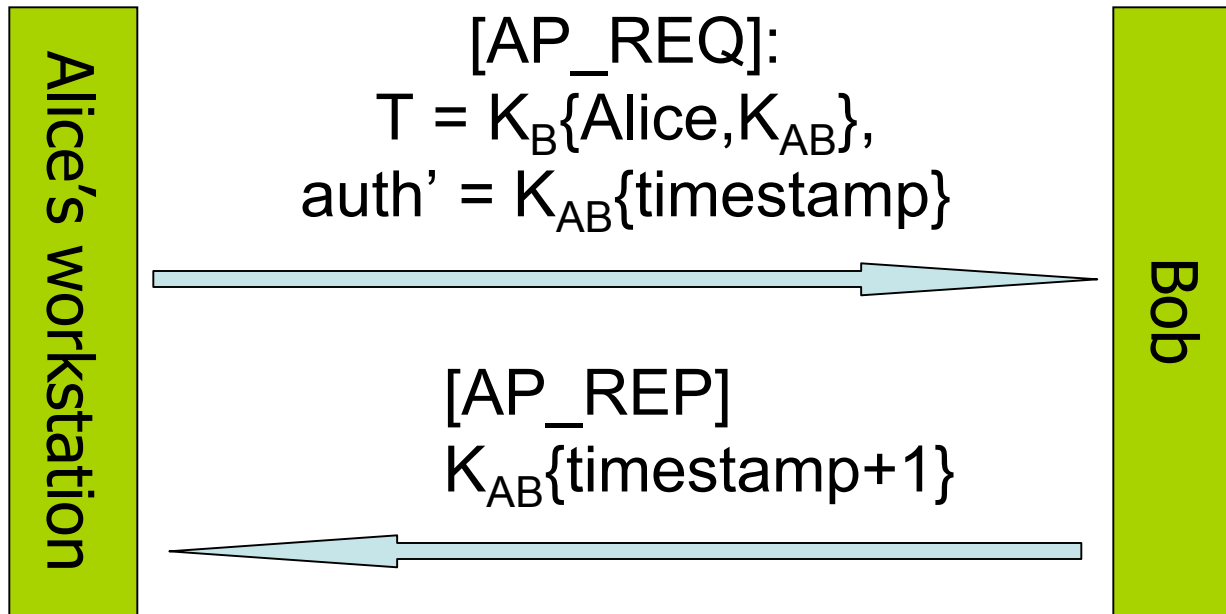
# First step: Get a TGT



# Getting a service ticket



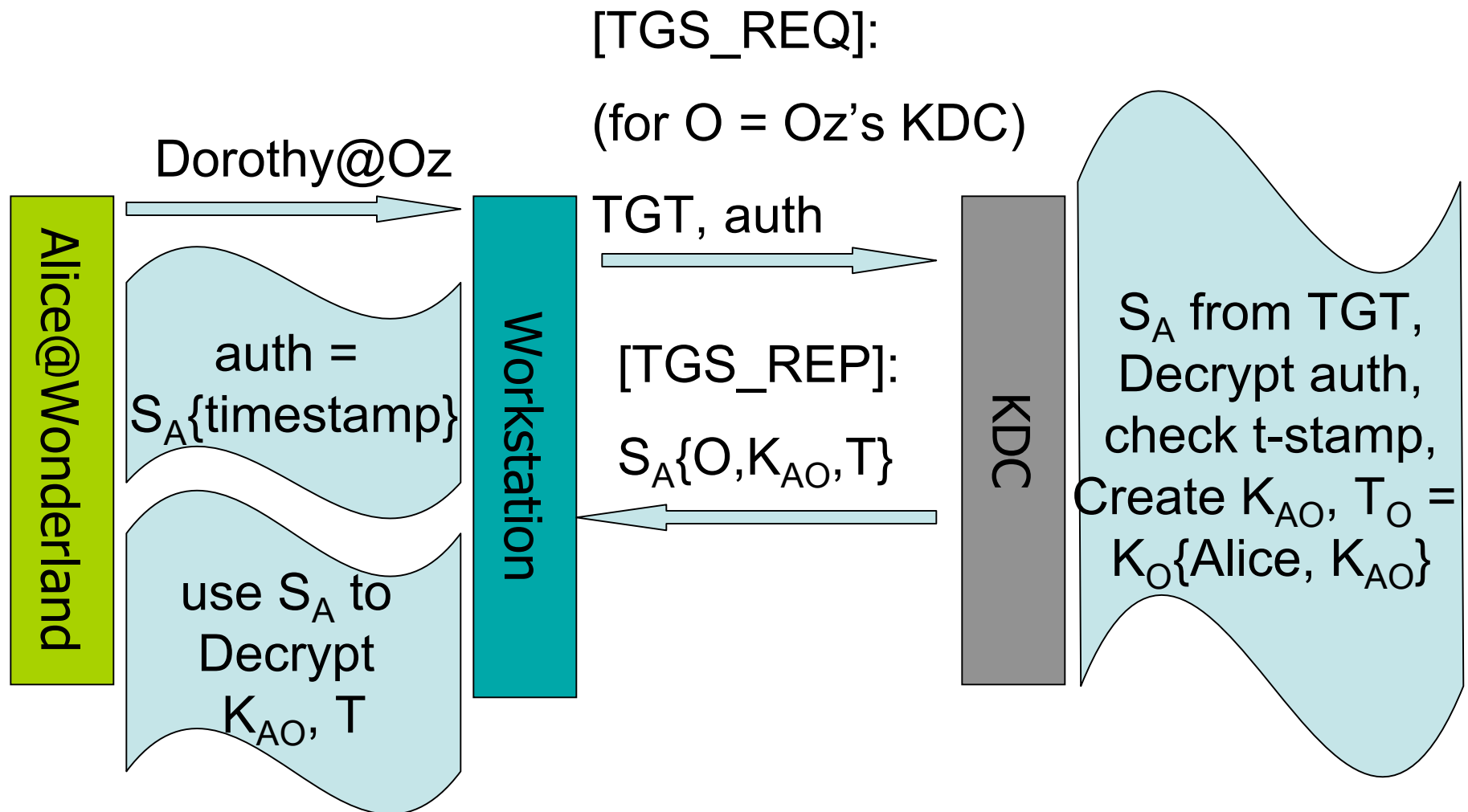
# Using the ticket



# Replicating KDCs

- To alleviate bottlenecks, replicate KDC:
  - One KDC is master database, to add or remove users, and change passwords
  - Other KDCs use database as read-only
- A master KDC establishes a *realm*.  
*Inter-realm* authentication supported:
  - $KDC_1$  registers  $KDC_2$  as a principal
  - $KDC_1$  enables other principals to access  $KDC_2$  as a kerberized service.

# Inter-realm authentication: 1



# Inter-realm authentication: 2

