

Network Authentication Standards:
Kerberos

- Kerberos designed at *MIT* and it is name of a 3 headed dog!
- It is a *secret key* based service for providing authentication in a network.
- Some *applications* that use Kerberos: *telnet, rsh and NFS*.

Master Keys and Session Keys:

- The KDC shares a secret key, called the *master key*, with each principle (each user and each resource). Alice's master key K_A is derived from her password.
- The workstation asks the KDC for a limited-lifetime *session key* S_A

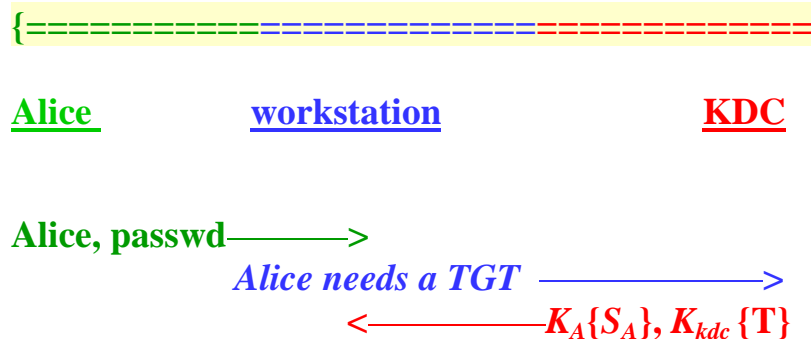
The KDC sends the workstation:

$K_A\{S_A\}$ and a *ticket-granting ticket* (TGT) $K_{kdc}\{T\}$

T contains: *Alice's name, S_A and expiration time.*
 K_{kdc} is the he KDC master key.

The workstation
forgets Alice's password K_A and
remembers S_A and the TGT.

This is illustrated as:



=====}

When Alice needs to talk to Bob (e.g., % rsh Bob)

- Her workstation sends the TGT to the KDC.
The KDC generates K_{AB} and send to the workstation:
 $S_A\{K_{AB}\}$ & a **ticket to Bob** = $K_B\{ "Alice", K_{AB}\}$
- Her workstation sends this ticket to Bob along with an **authenticator** $K_{AB}\{t\}$
where t is the current time to prove to Bob that she knows K_{AB}
(Kerberos allows up to *5 minutes skew* between clocks).
- Bob sends back $K_{AB}\{t+1\}$ to prove that he is indeed Bob
(since he must know K_B to find out K_{AB}).
- Thereafter, messages between Alice and Bob may be encrypted and integrity protected.

This is illustrated as:

