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.
- **Version 4** is simple and efficient, while **Version 5** has more functionality.
- Some *applications* that use Kerberos: *telnet, rsh, 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 and only remember $S_A$ and the TGT.

  This is illustrated as:

  ```
  Alice workstation KDC
  Alice, passwd──> Alice needs a TGT ───> K_A\{S_A\}, K_{kdc}\{T\}
  ```

**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:
  $S_A\{K_{AB}\}$ and a ticket to Bob = $K_B\{"Alice",K_{AB}\}$

- Her workstation sends this ticket to Bob along with an *authenticator* $K_{AB}\{Tmd\}$
where \( T_{md} \) is the time of day to prove to Bob that she knows \( K_{AB} \) (Kerberos allows up to 5 minutes skew between clocks).

- Bob sends back \( K_{AB} \{T_{md}+1\} \) to prove that he is indeed Bob
  (since he must knows \( K_B \) to find out \( K_{AB} \)).
- Thereafter, messages between Alice and Bob may be encrypted and integrity protected.

This is illustrated as:

```
Alice workstation KDC Bob
rsh Bob

Alice wants Bob, TGT 

<-------- S_A{"Bob", K_{AB}, ticket to Bob}

ticket to Bob = K_B{"Alice", K_{AB}, K_{AB}\{T_{md}\}}

<----------------------------- K_{AB}\{T_{md}+1\}
```

**Replicated KDC**

To avoid a single-point-of-failure and a performance-bottleneck of using one KDC, Kerberos keeps a master KDC and a number of slave KDCs. All updates are done at the master KDC database and periodically the slaves download the master database (the database consists of the principles names and master keys encrypted with the KDC master key). Most KDC operations are read-only and can be performed by any KDC replica.

**Key Version Numbers**

- If a non-human resource (e.g., a file server) changes his master key, and some one
  has a ticket for that resource with an old key, the ticket will no longer be valid
  (most tickets expire in about 21 hours). Kerberos assign to each key a version number and resources should remember
several versions of their own key. In tickets, key number version is sent so that it can be known which key to use.

- For **humans**, key version number is a problematic since Kerberos decided not to add further complexity to the user interface. If a person changes his master key and tried to login to a slave KDC before it has a chance to download the current key database, the person will not be able to login with new password!

**Encryption for Privacy and Integrity**

There is no standard mechanism for protecting both the *privacy* and *integrity* of a message with a single cryptographic pass. Therefore, Kerberos did a modified version of CBC called *Plain Cipher Block Chaining (PCBC)* as shown:

![Diagram of Encryption for Privacy and Integrity](image)

It has the property that modifying any \( c_i \) will result in garbling all blocks from \( m_i \) to the end. (Kerberos puts some recognizable data at the end, e.g., 1111).

**Encryption for Integrity only**

Kerberos V4 uses *checksum* function on the session key concatenated with the message, but in V5 they use one of the standard *message digest* functions.

**Network Layer Addresses in Tickets**
• In V4, when Alice requests a TGT or a ticket to Bob, the KDC puts her 32-bit IPv4 address inside the ticket!

• This restricts Alice to "legitimately" allow something to act on her behave, thus V4 does not allow delegation.

**Delegation**

• V5 explicitly allows delegation by allowing Alice to ask for a TGT ticket with a network layer address different from hers (also she can ask for multiple or no address).

• She can then give this ticket to Bob (or anyone that matches the network address) along with the corresponding session key.

• Bob can then uses this TGT to obtain tickets for the needed resources.

**Ticket Lifetime**

In V4 tickets is valid for 21 hours.
In V5, it can be issued for any duration and it can be renewable and postdated.