Plan of Talk

- **Yesterday-**
  - **Mutual Trust using Symmetric key techniques:**Needham-Schroeder Protocol
- **Today**
  - **Example: Kerberos**
  - Web Security
  - **Basics of SSL**
- NOTE: Please study the detailed slides provided to you.

Kerberos

- **What is Kerberos?**
  - **is an authentication server developed as a part Project Athena, MIT**
  - **Kerberos provides centralised private-key third-party authentication in a distributed network**
- What problem was Kerberos designed to address?

Threats

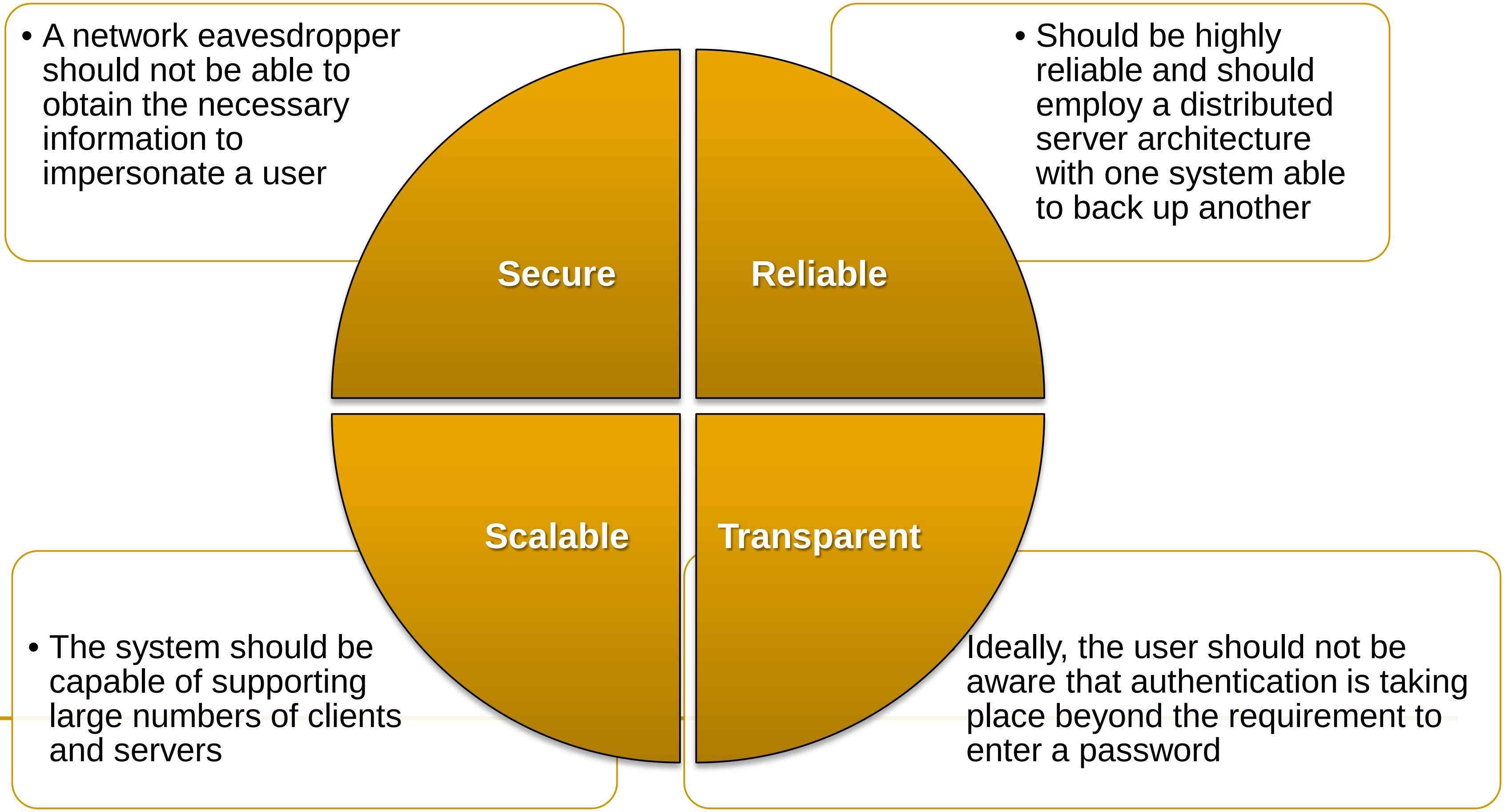
- **What are three threats associated with user authentication over a network or Internet?**
  - **Masquerading: Gain access to a particular workstation and pretend to be someone.**
  - **Adversary may change the network address for impersonation.**
  - **Eavesdrop communication for other malicious activities (replay etc).**

Authentication in a distributed environment

- **List three approaches to secure user authentication in a distributed environment**
  - **Based on Each Individual workstation assuing User Identification.**
  - **Client Systems authenticate to servers**
  - **User to prove identity for each service invoked.**

Kerberos Requirements

- The first published report on Kerberos listed the following requirements:



Secure

- A network eavesdropper should not be able to obtain the necessary information to impersonate a user

Reliable

- Should be highly reliable and should employ a distributed server architecture with one system able to back up another

Scalable

- The system should be capable of supporting large numbers of clients and servers

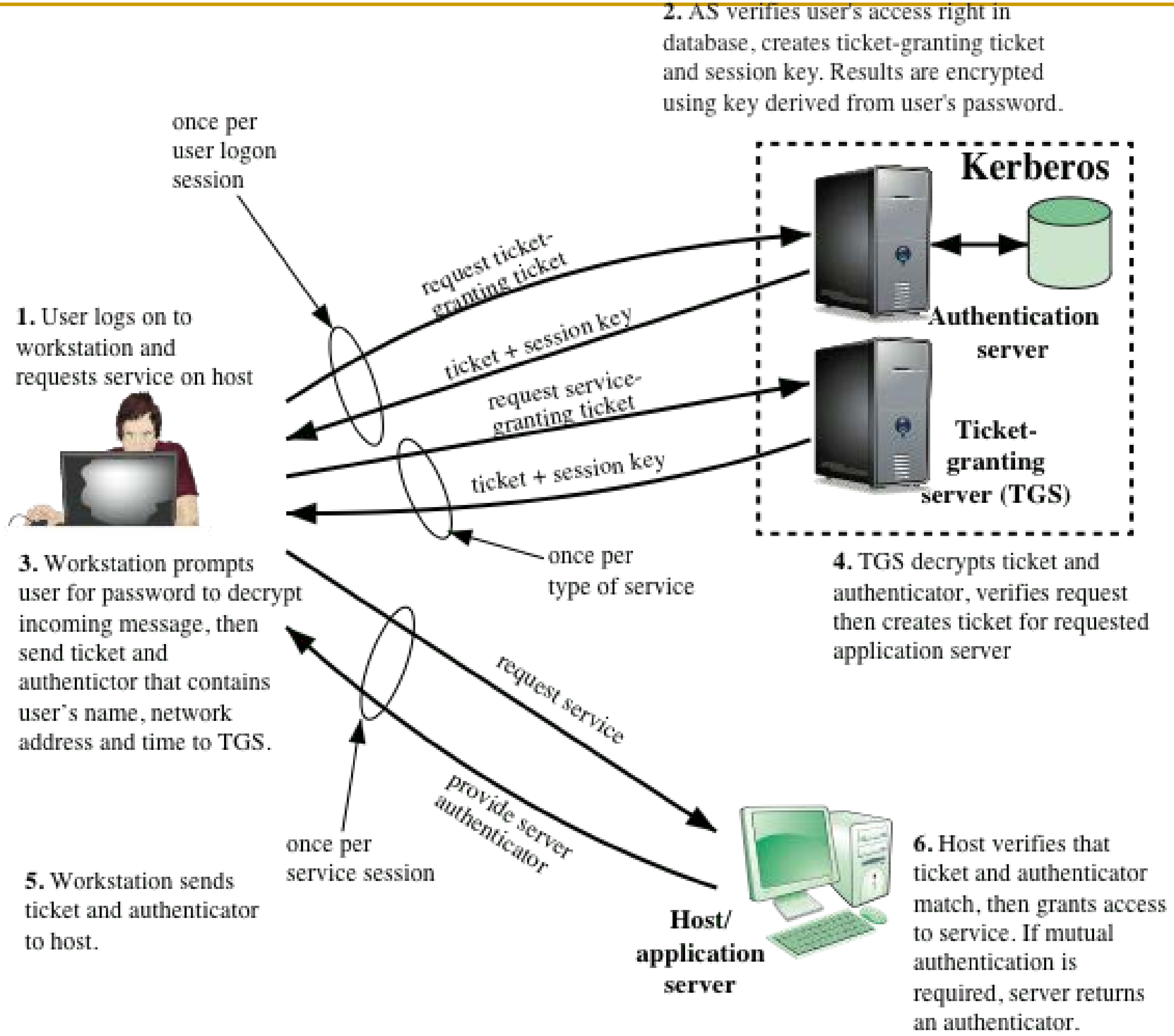
Transparent

Ideally, the user should not be aware that authentication is taking place beyond the requirement to enter a password

Kerberos Version 4: Overview

- Makes use of DES to provide the authentication service
- Authentication server (AS)
  - Knows the passwords of all users and stores these in a centralized database
  - Shares a unique secret key with each server
- Ticket
  - Created once the AS accepts the user as authentic; contains the user's ID and network address and the server's ID
  - Encrypted using the secret key shared by the AS and the server

- Ticket-granting server (TGS)
  - Issues tickets to users who have been authenticated to AS
  - Each time the user requires access to a new service the client applies to the TGS using the ticket to authenticate itself
  - The TGS then grants a ticket for the particular service
  - The client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested



2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

1. User logs on to workstation and requests service on host

3. Workstation prompts user for password to decrypt incoming message, then send ticket and authentictor that contains user's name, network address and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request then creates ticket for requested application server

5. Workstation sends ticket and authenticator to host.

6. Host verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

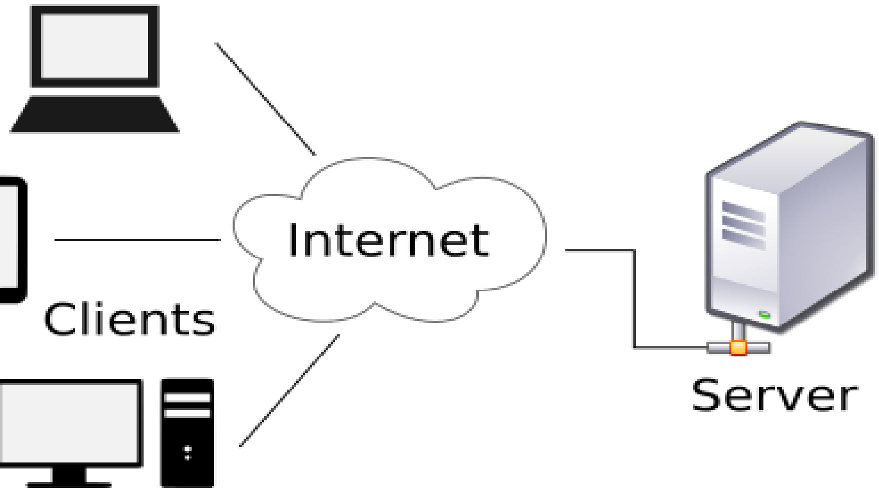
Figure 15.1 Overview of Kerberos

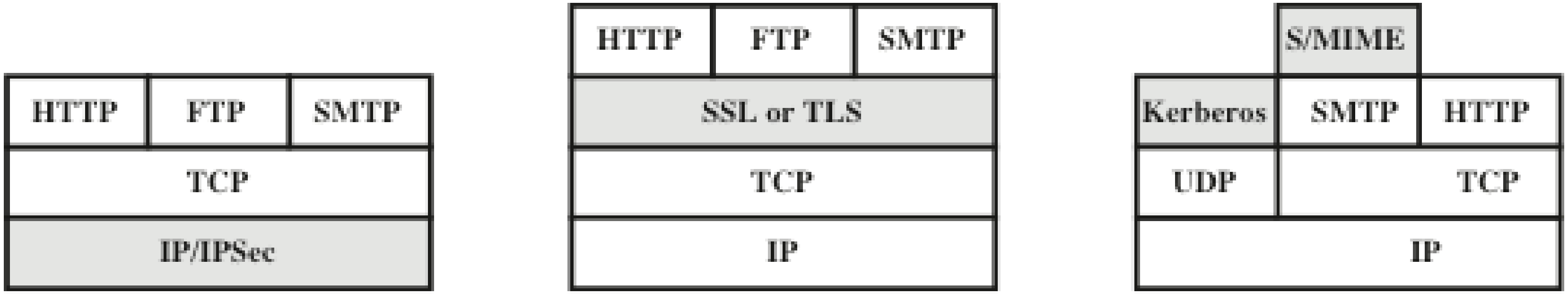
Kerberos Realms

- **A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires the following:**
  - **1. The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.**
  - **2. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.**

Plan of Talk

- **Transport Layer Security**
- **SSL**




HTTP | FTP | SMTP
TCP
IP/IPSec

(a) Network Level

HTTP | FTP | SMTP
SSL or TLS
TCP
IP

(b) Transport Level

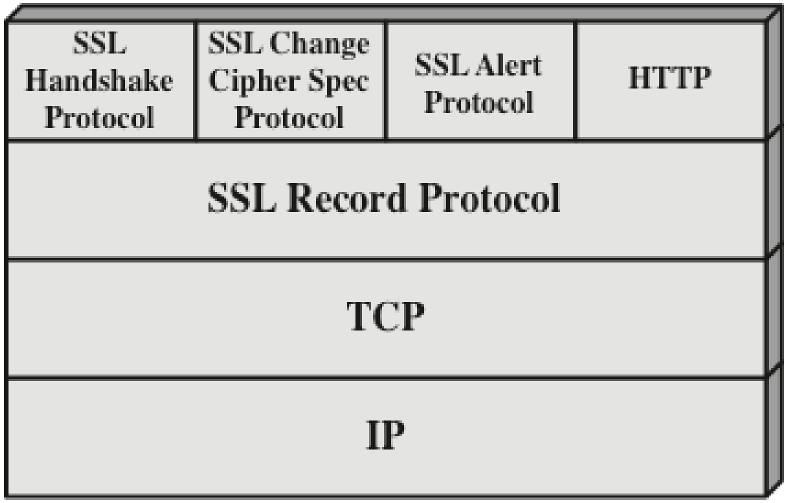
S/MIME
Kerberos | SMTP | HTTP
UDP | TCP
IP

(c) Application Level

Secure Sockets Layer (SSL)

- One of the most widely used security services.
- A general purpose service implemented as a set of protocols that rely on TCP
  - Could be provided as part of the underlying protocol suite and therefore be transparent to applications
  - Can be embedded in specific packages

Two issues:
NETWORK
CRYPTO ALGORITHMS



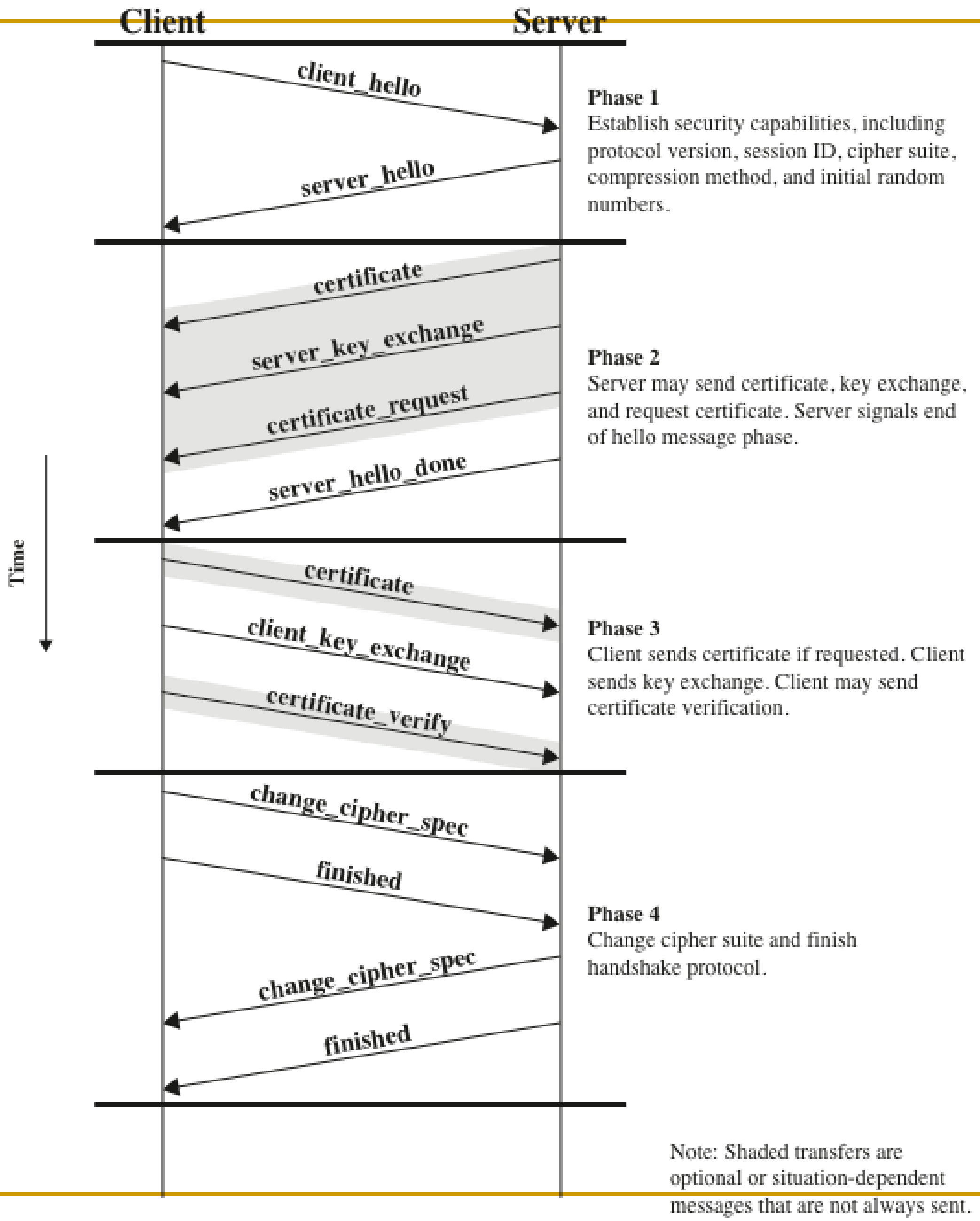
SSL Handshake Protocol | SSL Change Cipher Spec Protocol | SSL Alert Protocol | HTTP

SSL Record Protocol

TCP

IP

Figure 17.2 SSL Protocol Stack



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Figure 17.6 Handshake Protocol Action

Cryptographic Computations

- Two further items are of interest:
  - The creation of a shared master secret by means of the key exchange
    - The shared master secret is a one-time 48-byte value generated for this session by means of secure key exchange
  - The generation of cryptographic parameters from the master secret
    - CipherSpecs require a client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV which are generated from the master secret in that order
      - These parameters are generated from the master secret by hashing the master secret into a sequence of secure bytes of sufficient length for all needed parameters

Summary

- Kerberos
- Web Security

  - Please read ApplicationKerberos-Notes.pdf
  - WebSecurity-SSL-Notes on LMS for more details.